*Bachelor Arbeit*

# Analyse und Entwurf von Kalendersystemen unter besonderer Berücksichtigung von Anwendungsfällen und Synchronisation

**Timo Trautmann**
Matrikel-Nummer: 643836

Referent: Prof. Dr.-Ing. Bernhard Kreling
Korreferent: Prof. Dr. Herwig Meyer

*Fachhochschule Darmstadt*
*Fachbereich Informatik*

*Mai 2005*

# Erklärung

Hiermit versichere ich, dass ich meine Bachelor-Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Darmstadt, den 10. Mai 2005 ___________________________________

Timo Trautmann

# Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen und allen Personen danken, die mir bei meiner Bachelor Arbeit geholfen haben.

In erster Linie möchte ich mich bei meinem Freund Jörg Steinmetz bedanken, der die gesamte Arbeit Korrektur gelesen hat. Des Weiteren danke ich ihm für den ein oder anderen Screenshot einer Kalenderapplikation, die ich nicht zur Verfügung hatte. Abschließend möchte ich ihm noch für den ein oder anderen hilfreichen Weblink danken.

Mein Dank geht auch an Elke Trautmann, die die gesamte Arbeit ebenfalls auf Rechtschreib- und Tippfehler durchgelesen hat. Danken möchte ich des weiteren Aboulkacime Chebak, der ebenfalls einen Blick in die Arbeit geworfen hat.

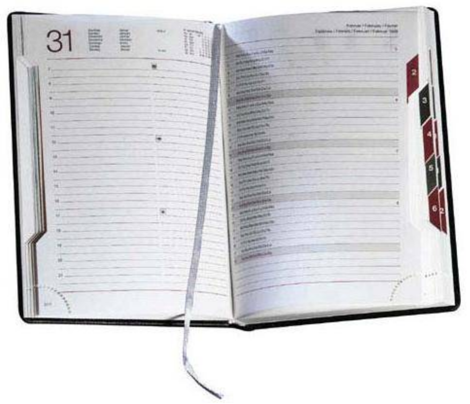
Ein weiterer besonderer Dank geht an meinen Referenten Bernhard Kreling, der mir bei Fragen sowohl persönlich als auch per E-Mail immer mit Rat und Tat zur Seite stand. Ihm auch meinen Dank für das Durchlesen dieser Arbeit.

In diesem Zusammenhang möchte ich auch meinem Korreferenten Herwig Meyer recht herzlich für das Lesen der Arbeit danken.

## Inhaltsverzeichnis

# 1 Einführung

## 1.1 Allgemeine Einführung


*Abb. 1 - Der klassische Terminplaner*

Längst ist die Zeit vorbei, zu der man seine Termine noch in die guten alten Terminbücher eintragen musste (Abbildung rechts). Im Zeitalter des Computers verwaltet man seine Termine auch auf digitale Art und Weise. Kein lästiges Blättern mehr, um sich alle Termine der kommenden Woche oder des ganzen Monats anzuschauen. In Kalendersystemen ist man meist nur einen Mausklick von solchen Übersichten entfernt. Wöchentliche Termine muss man nicht mehr jede Woche in seinen Kalender schreiben. In die modernen Systeme gibt man den Termin nur noch einmal mit dem Vermerk wöchentlich ein und die Software trägt den Termin jede Woche automatisch ein. Wie oft hat man schon einen Termin vergessen? Dank der neuen Software auch kein Problem mehr. Man kann sich an jeden Terminen erinnern lassen. Wenn man seine Mitarbeiter über eine Besprechung informieren möchte, muss man nicht mehr umständlich jeden einzeln ansprechen. Einige Terminplanersysteme unterstützen beim Anlegen des Termins auch den Terminversand per E-Mail an andere Personen. Manche Systeme sind sogar schon so weit, dass der Termin beim Empfänger der E-Mail automatisch in seinen Terminplaner eingetragen wird. Ist ein Dokument an die Beteiligten einer Besprechung auszuhändigen, wie zum Beispiel eine Agenda, ist das heute auch sehr einfach. In einigen Terminplanersystemen kann man Dateien an die Termine anhängen. Bei Benachrichtigung der Beteiligten werden diese Dateien einfach an eine E-Mail angehängt, die an die Beteiligten versandt wird.

Die Kalendersysteme haben den lokalen Einzelplatzrechner oder Laptop längst verlassen. Sie sind fast überall zu finden, wie zum Beispiel auf PDAs. Ja sogar das Handy wird schon zur mobilen Organisationszentrale. Längst nutzt der Benutzer nicht nur ein Kalendersystem, um seine Termine komfortabel zu verwalten. Egal wo man sich befindet, immer hat man seine aktuellen Termine zur Verfügung.

Daher wird in der heutigen Zeit die Synchronisation von Daten zwischen Kalender-systemen immer wichtiger. Schließlich will man seine Termine, die man auf dem PDA oder dem Handy eingegeben hat, nicht mühsam auf seinen Einzelplatzrechner abtippen oder umgekehrt.

## 1.2 Motivation

Bei dieser Fülle verliert man schnell den Überblick über die vielen verschiedenen Arten von Terminplanersystemen. Diese Abschlussarbeit soll in drei größeren Abschnitten etwas Licht ins Dunkel bringen: Es soll dargestellt werden, aus welchen Komponenten Terminplanersysteme bestehen, welche wesentlichen Funktionen sie beinhalten und auf welchen Systemumgebungen sie zu finden sind.

In einem weiteren größeren Abschnitt soll dann Bezug dazu genommen werden, welche Technologien es gibt, Termindaten zwischen verschiedenen Terminplanern zu synchronisieren.

In einem dritten und letzten Abschnitt soll dann auf das Kalendersystem näher eingegangen werden, das ich in meiner Praxisphase entwickelt habe. Dieses Kalendersystem soll dann in Bezug zu den bereits vorhandenen Technologien gesetzt werden.

Die theoretischen Grundlagen werden immer wieder durch praktische Beispiele und Screenshots veranschaulicht.

## 1.3 Hintergrund

In meiner vorausgegangen Praxisphase von Anfang Oktober 2004 bis Mitte Februar 2005 an der Fachhochschule Darmstadt habe ich bei Cross-Consulting an der Anwendung Cross-Desktop gearbeitet. Cross-Consulting ist eine Beratungsgesellschaft, die sich unter anderem mit dem Thema Content-Management beschäftigt. Der Firmensitz ist im Modautal, Ortsteil Allertshofen (Odenwald). Dies ist in der Nähe von Darmstadt. Weitere Informationen zum Unternehmen kann man unter http://www.cross-consulting.com in Erfahrung bringen.

*Abb. 2 - Cross-Desktop Logo*

Der Cross-Desktop ist eine webbasierte Applikation, mit der man seine E-Mails verwalten kann, ähnlich einer Anwendung, wie Outlook oder Mozilla Sunbird. (Outlook und Sunbird sind auf lokalen Einzelplatzrechnern zu finden.) Außerdem ist es möglich, seine Dateiablage webbasiert zu managen und RSS-Feeds zu verwalten. Die Anwendung integriert Dateimanagement und E-Mail in nur einer Applikation. In Kapitel 6 wird noch näher auf diese Anwendung eingegangen. Meine Aufgabe war es, einen webbasierten Terminplaner für die Webapplikation Cross-Desktop zu realisieren. Die Anwendung wurde serverseitig mit Perl und XML realisiert und clientseitig mit HTML, CSS und JavaScript. Im Nachfolgenden wird dieser Terminplaner Cross-Calendar genannt. Die Applikation beinhaltet alle wesentlichen Bestandteile eines Kalendersystems, die im nachfolgenden Kapitel zwei beschrieben werden. In Kapitel 6 wird ebenfalls näher auf den Cross-Calendar eingegangen.

Da ich in meiner Praxisphase nur einen webbasierten Kalender entwickelt habe und dabei auch keinen Bezug zu bereits existierenden Standards für Kalendersysteme hergestellt habe, möchte ich dies mit dieser Arbeit nachholen. Ich möchte einen Einblick geben, welche Standards es gibt, wo Kalendersysteme zu finden sind und wann sie zum Einsatz kommen. Abschließend sollen dann die verschiedenen Technologien verglichen und diskutiert werden. [CC05] [MH05]

## 1.4 Kurzübersicht über die folgenden Kapitel

Bevor wir in das zweite Kapitel “Typische Komponenten eines Kalendersystems“ einsteigen, soll an dieser Stelle zur besseren Orientierung eine kurze Beschreibung der einzelnen nachfolgenden Kapitel gegeben werden:

Kapitel zwei stellt zunächst die typischen Komponenten eines Kalendersystems dar. Im Wesentlichen beinhaltet solch ein System die Funktionen Termine eintragen, Termine anzeigen, Terminerinnerung, sowie Im- und Exportfunktionen von Termindaten. Termine

lassen sich dann noch zusätzlich in einmalige Termine, periodische Termine, und To-Do's einteilen. Diese Termine lassen sich auch auf verschiedene Arten visualisieren. Es werden die Darstellungsarten Tages-, Wochen- und Monatsansicht dargestellt und ein kurzer Einblick gegeben, welche weiteren weniger verbreiteten Ansichtsformen es gibt. Im letzten Abschnitt des Kapitels werden dann Beispiele für Funktionalitäten in Kalendersystemen gegeben, die nicht in jedem Kalendersystem zu finden sind. Da es sehr viele solcher Funktionen gibt, kann auch hier nur ein kleiner Einblick gegeben werden.

Um besser verstehen zu können, warum es auf so vielen verschiedenen Plattformen Kalendersysteme gibt, soll Kapitel zwei erläutern, in welchen Situationen Menschen mit Kalendersystemen arbeiten. Also zum Beispiel im Büro oder auf Reisen und welche Anforderungen in dieser Situation an das System gestellt werden.

Dies stellt dann die Überleitung zum nachfolgenden Kapitel dar, das zeigt, auf welchen Plattformen Kalendersysteme zu finden sind. Also zum Beispiel auf Handys oder dem lokalen Einzelplatzrechner. Zu jeder Plattform wird dann noch kurz mindestens ein Kalendersystem kurz vorgestellt. Abschließend werden die Vor- und Nachteile dieser Plattformen herausgearbeitet.

Aufgrund dieser Fülle verschiedener Software und verschiedenen Plattformen wird schnell klar, dass Kalendersysteme meist in Kombination zur Anwendung kommen. Kapitel fünf soll einen kleinen Einblick geben, welche Standards zur Synchronisation es gibt, mit denen man seine Termindaten zwischen den Anwendungen abgleichen kann. Das Kapitel beinhaltet im Wesentlichen drei Technologien: iCalendar, CalDAV und SyncML. Diese Technologien werden an geeigneter Stelle mit Beispielen näher erläutert.

In Kapitel sechs möchte ich dann etwas näher auf meine Tätigkeit in der Praxisphase eingehen. Ich möchte den Cross-Calendar vorstellen, sowie sein Konzept zur Realisierung. Da der Cross-Calendar nur ein Bestandteil des Cross-Desktops ist, möchte ich auch eine kurze Einführung in den Cross-Desktop geben, damit man meine Arbeit in der Praxisphase besser verstehen kann.

Kapitel sieben soll dann den Cross-Calendar in die beschriebenen Plattformen einordnen und Vor- und Nachteile zu anderen Kalendersystemen, die auf dem Markt zu finden sind, beschreiben. Außerdem möchte ich noch einen Ansatz geben, wie man den Synchronisationsstandard iCalendar in den Cross-Calendar implementieren könnte. Des Weiteren möchte ich kurz darstellen, was zu tun wäre, um die Termine in einer Datenbank statt in XML zu speichern und wo darin Vor- und Nachteile liegen.

Das Kapitel acht soll den Abschluss dieser Arbeit darstellen. Es soll eine Art kurze Stellungnahme zu den zuvor beschriebenen Kapiteln sein. Hier soll dargestellt werden, wo Stärken und Schwächen in Kalendersystemen zu finden sind und in welchen Bereichen noch ein Entwicklungsbedarf besteht.

Im Anhang sind alle verwendeten Literaturstellen zu finden, die bei dieser Arbeit verwendet wurden. Auch ein Quellenverzeichnis der Abbildungen ist hier aufgelistet.

# 2 Typische Komponenten eines Kalendersystems

In diesem Abschnitt soll geklärt werden, aus welchen Bestandteilen ein Kalendersystem besteht, um später näher darauf eingehen zu können, welche Komponente der Benutzer in welcher Situation benötigt und welche nicht.

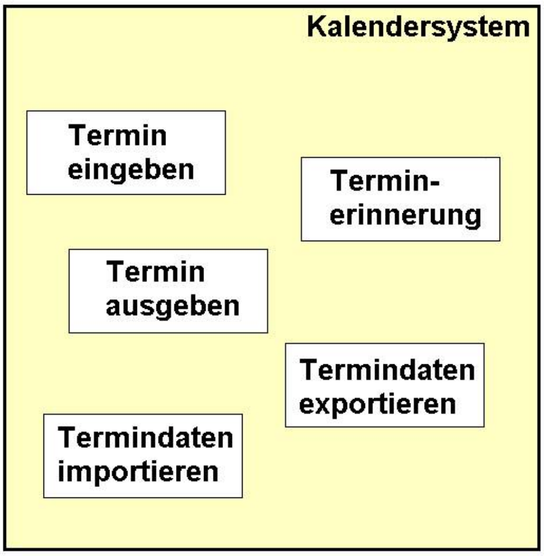


*Abb. 3 - Komponenten eines Kalendersystems*

Das rechts stehende Diagramm zeigt typische Komponenten solcher Systeme. Man kann Termine verschiedener Typen eingeben. Diese Termine kann man dann später auf vielfältigste Art und Weise wieder am Bildschirm ausgeben. Damit man keinen Termin mehr vergisst, stellen die meisten Systeme auch eine Funktion zur Terminerinnerung zur Verfügung. Um seine Termindaten zu sichern oder auf anderen Systemen zu verwenden, stellen die Systeme auch Komponenten zum Im- und Export von Daten zur Verfügung. Die folgenden Abschnitte dieses Kapitels sollen diese Komponenten anhand praktischer Beispiele näher beschreiben und die Arten der Termineingabe und Terminausgabe näher klassifizieren.

## 2.1 Termine

Ein wesentlicher Bestandteil von Kalendern ist die Möglichkeit, Termine einzutragen. Man unterscheidet zwischen einmaligen und periodischen Terminen, sowie To-Do's. Unter einmaligen Terminen versteht man einen Termin, der nur einmal statt findet. Unter periodischen Terminen versteht man Termine, die zum Beispiel einmal pro Woche statt finden. Unter To-Do's versteht man Aufgaben, denen kein konkreter Zeitpunkt zugeordnet wird. Zum Beispiel möchte man eine Bank-Überweisung tätigen. Ob man das aber Vormittags oder Nachmittags erledigt, ist egal. Die nachfolgenden drei Abschnitte sollen diese drei Arten von Terminen etwas näher erläutern.

### 2.1.1 Einmalige Termine

Einmalige Termine ist die Terminform die am Häufigsten in Kalendersystemen auftritt. Der Termin findet genau einmal zu einem festgelegten Zeitpunkt statt und hat in der Regel auch eine festgelegte Zeitspanne. Der Benutzer kann seinem Termin einen Titel geben und den Ort bestimmen, an dem dieser Termin statt findet. Danach gibt man ein Startdatum und ein Enddatum mit zugehöriger Uhrzeit ein. Termine über mehrere Tage sind hierbei auch kein Problem. Des Weiteren man eine ausführliche Beschreibung zu diesem Termin eingeben. Außerdem kann noch festlegen, in welchem Zeitraum man an diesen Termin erinnert werden möchte und ob dieser Termin privater oder geschäftlicher

Natur ist. Auf die Terminerinnerung wird in einem gesonderten Abschnitt genauer eingegangen. In den meisten Systemen kann der Benutzer jedem Termin eine Kategorie zuordnen. Normalerweise kann er auch selbst Kategorien anlegen und verwalten. Als Beispiele für Kategorien wären Geburtstag, Reise, Urlaub, Geschäftsessen oder ähnliches zu nennen. Außerdem kann man vermerken, wie der aktuelle Status des Termins ist. Es kann bestimmt werden, ob der Termin festgelegt, abgesagt oder nur geplant ist. In den meisten Systemen ist es möglich, jedem Termin eine Priorität zuzuweisen. Eine Priorität gibt an, wie wichtig der Termin ist.

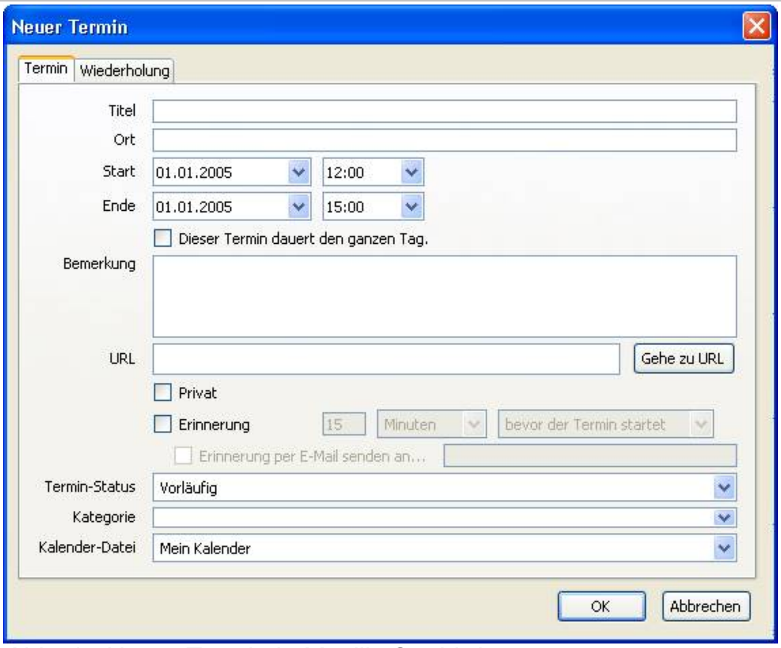


*Abb. 4 - Neuer Termin in Mozilla Sunbird*

Dies sind die wesentlichen Bestandteile eines einmaligen Termins. Die einzelnen Kalendersysteme haben manchmal noch spezielle Funktionen. An dieser Stelle ist ein Screenshot aus Mozilla Sunbird zu sehen. Neben den bereits beschriebenen Funktionen erleichtert eine Checkbox ganztägige Termine einzutragen. Auch hat hier der Benutzer die Möglichkeit jedem Termin eine URL zuzuweisen und sich per E-Mail an den Termin erinnern zu lassen. In Microsoft Outlook hingegen hat man als weitere Funktion die Möglichkeit, den Termin per E-Mail an andere Benutzer zu versenden, die an diesem Treffen beteiligt sind. Des Weiteren kann man in Microsoft Outlook an den Termin auch Dateien anhängen. Dies ist zum Beispiel dann sinnvoll, wenn es sich um ein geschäftliches Treffen handelt und eine Agenda oder sonstige Dokumente mit diesem Treffen verbunden sind. [SUN02] [OUT05]

## 2.1.2 Periodische Termine

Eine weiterer Termintyp sind die periodischen Termine. Nicht alle Termine finden nur ein einziges Mal statt. Als Beispiele kann man jährliche Geburtstage oder wöchentliche Besprechungen nennen. Die verschiedenen Anwendungen ermöglichen es daher, periodische Termine einzutragen, so dass man nicht den gleichen Termin immer und immer wieder mühselig eintragen muss. Ein periodischer Termin hat die gleichen Komponenten wie die unter 2.1.1 beschriebenen Termine. Zusätzlich enthält der Termin jedoch noch Informationen über die Periode.

Man unterscheidet eine tägliche, wöchentliche, monatliche und jährliche Periode. Bei der täglichen Periode kann man bestimmen, alle wie viele Tage sich der Termin wiederholt. Zum Beispiel wiederholt sich ein Termin alle drei Tage. Bei der wöchentlichen Periode kann man auswählen, an welchen Wochentagen der Termin statt findet. Zusätzlich kann man noch vermerken, alle wie viele Wochen diese Wiederholung erfolgt. Kombinationen wie beispielsweise alle zwei Wochen am Montag und Mittwoch sind somit möglich. Die

monatliche Periode stellt zwei Arten der Wiederholung zur Verfügung. Einmal die Wiederholung nach Wochentag und einmal die Wiederholung nach Monatstag. Wiederholung nach Monatstag bedeutet, dass der Termin immer zum Beispiel am 15. eines Monats wiederholt wird. Wiederholung nach Wochentag legt hingegen die Woche und den zugehörigen Wochentag im Monat fest auf dem der Termin liegt. So kann man zum Beispiel ein Treffen jeden 2. Mittwoch im Monat haben. Auch hier besteht die Möglichkeit, dass dieser Termin nicht jeden Monat statt findet, sondern zum Beispiel nur alle vier Monate. Zuletzt gibt es noch die jährliche Periode. Ein Termin findet einmal im Jahr statt. Beispielsweise hat Hugo am 25.12. Geburtstag. Auch bei der jährlichen Periode ist man nicht daran gebunden, dass der Termin einmal pro Jahr statt findet. Es ist kein Problem einzustellen, dass der Termin zum Beispiel nur alle fünf Jahre statt findet.

Dann stellt sich nur noch die Frage, wie lange diese Periode dauern soll. Hier hat der Benutzer drei Möglichkeiten: Bei Möglichkeit eins geht der Termin bis ins unendliche in die Zukunft. Bei einer weiteren Möglichkeit gibt der Benutzer die Anzahl an, wie oft der Termin statt finden soll. Zum Beispiel zehn mal. Er endet dann nach zehn Wiederholungen. Bei der dritten und letzten Möglichkeit gibt der Benutzer einfach das Datum an, an dem der Termin das letzte Mal statt findet. Zum Beispiel endet ein täglicher Termin am 31.12.2005.

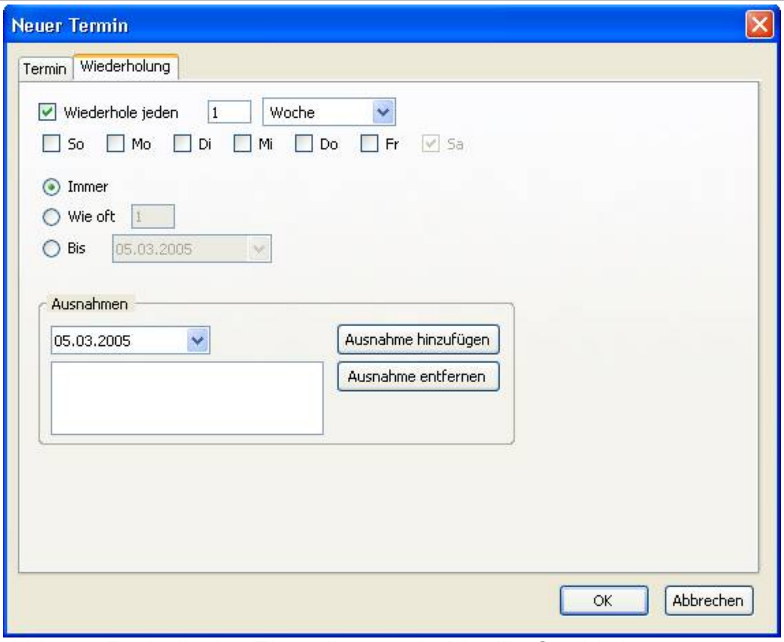


*Abb. 5 - Periodische Einstellungen in Mozilla Sunbird*

Von System zu System variieren die oben beschriebenen Optionen etwas. Sie beinhalten im Wesentlichen jedoch diese Möglichkeiten. In Mozilla Sunbird hat der Benutzer als zusätzliche Extrafunktion gleich beim Eintragen die Möglichkeit, verschiedene Datumsangaben einzugeben, an denen der Termin normalerweise statt finden würde, in diesen Fällen aber ausfällt. [SUN02]

### 2.1.3 To-Do's

Neben Terminen, die immer einen festen Beginn und ein festes Ende haben, kann man in Terminplanern auch Aufgaben erstellen, so genannte To-Do's. Diese Aufgaben haben typischerweise keinen festen zeitlichen Rahmen. Zum Beispiel ist eine Rechnung per Überweisung zu zahlen. Ob diese Zahlung morgens, mittags oder abends vorgenommen wird, ist egal. To-Do's haben, wie Termine auch, einen Titel und eine Beschreibung. Wenn diese Aufgabe außer Haus erledigt werden soll, kann auch ein Ort angegeben werden. Der Benutzer kann dieser Aufgabe, wie bei Terminen auch, einer Kategorie zuweisen und sie als privat markieren. Wichtigen Aufgaben kann eine höhere Priorität zugewiesen werden, als weniger wichtigen Aufgaben. Zum Beispiel ist eine geschäftliche Transaktion wichtiger, als ein Artikel in einer Zeitschrift, den man bei Gelegenheit noch lesen möchte. Bei umfangreicheren Aufgaben kann durch eine prozentuale Angabe vermerkt werden,

wie weit die Aufgabe bereits erledigt wurde. Da Aufgaben in der Regel keinen festen zeitlichen Rahmen haben, kann man sich auch nicht an Diese erinnern lassen.

Dennoch gibt es die Möglichkeit, einer Aufgabe einen zeitlichen Rahmen zuzuweisen, falls dies erforderlich ist. Man kann den Start und das Fälligkeitsdatum der Aufgabe vermerken. Dann ist es auch möglich, sich an den Beginn bzw. an das Ende der Aufgabe erinnern zu lassen. Wenn ein zeitlicher Rahmen festgelegt wurde, kann man den To-Do's periodische Eigenschaften, wie in Kapitel 2.1.2 beschrieben, zuweisen. Das heißt: Auch To-Do's können sich wiederholen. Zum Beispiel möchte man jeden Tag seine E-Mails überprüfen. Der genaue Zeitpunkt der Überprüfung ist dann nicht so wichtig. Wichtig ist nur, dass es in der Zeit von ... bis ... geschieht und das eben jeden Tag. [KOG05]

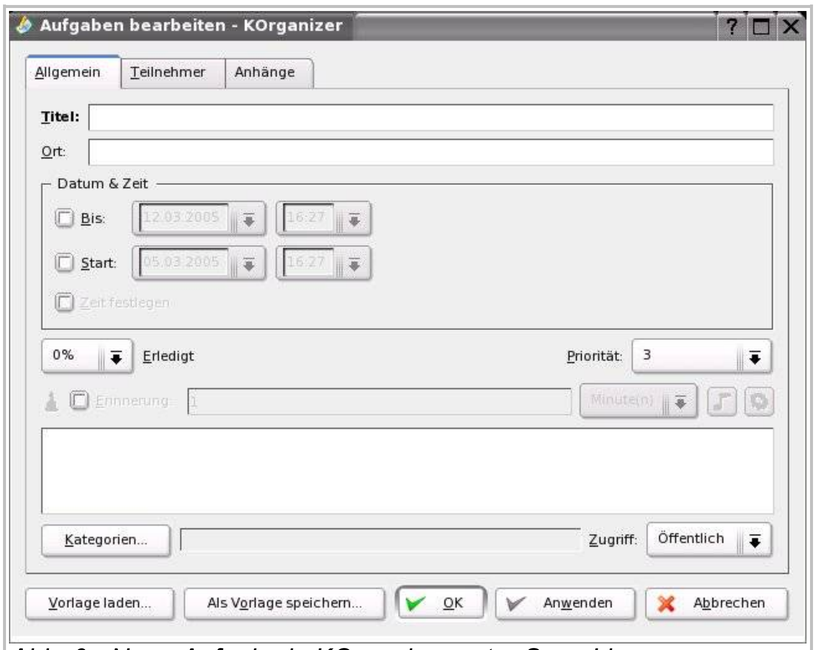


*Abb. 6 - Neue Aufgabe in KOrganizer unter Suse Linux*

## 2.2 Terminerinnerung

In den vergangenen Abschnitten war des öfteren die Rede von der Möglichkeit, sich an den Termin erinnern zu lassen. In diesem Abschnitt wollen wir uns diese Funktion etwas näher anschauen. In der Regel hat man nicht nur die Möglichkeit sich über den Beginn des Termins informieren zu lassen, sondern auch über dessen Ende. Ist die Zeit der Erinnerung gekommen, ertönt meist ein akustisches Signal und es erscheint ein kleines Fenster in dem auf den Termin hin gewiesen wird. Sind zu diesem Zeitpunkt mehrere Erinnerungen fällig, erscheinen in diesem Fenster auch mehrere Einträge an die erinnert werden soll.

In diesem Fenster hat man dann auch noch die Möglichkeit eine Zeitspanne einzugeben, nach der man erneut an diesen Termin erinnert werden möchte.

Als kleines Beispiel ist rechts ein Erinnerungsfenster aus Microsoft Outlook zu sehen. Man kann sich den Termin anzeigen lassen und unten eine erneute Erinnerung anfordern. [OUT05]

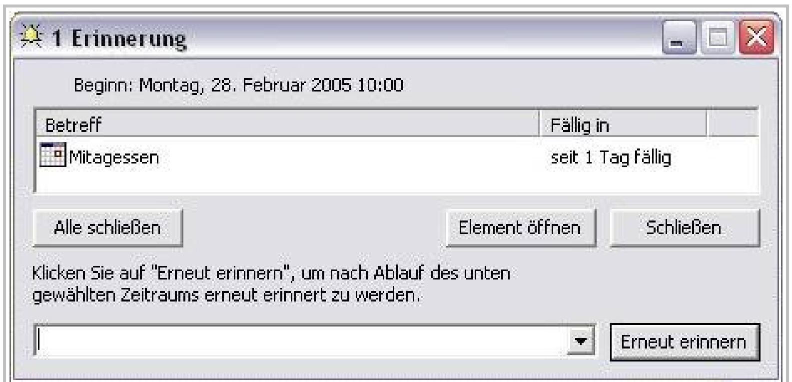


*Abb. 7 - Terminerinnerungsfenster in Microsoft Outlook*

## 2.3 Ansichten

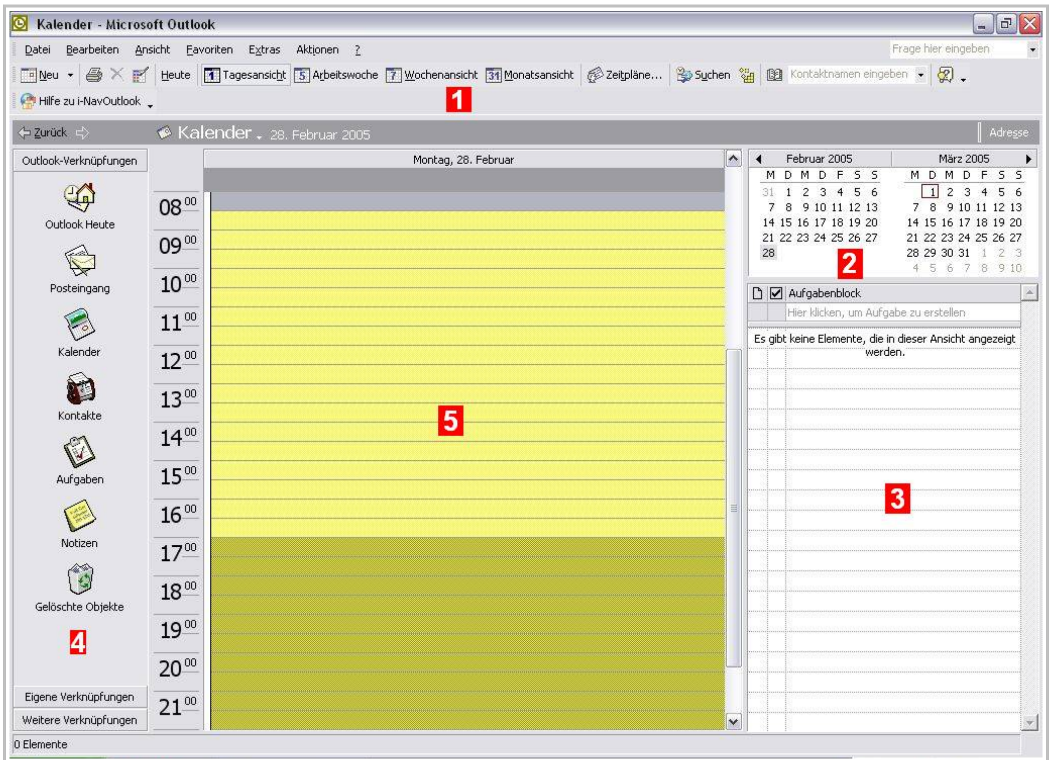



*Abb. 8 - Oberfläche des Terminplaners in Microsoft Outlook*

Dieser Abschnitt soll Aufschluss darüber geben, welche verschiedenen Arten es gibt, die Termine auf dem Bildschirm zu visualisieren. Im Wesentlichen unterscheidet man zwischen der Tages-, Wochen- und Monatsansicht. Diese Arten werden in den folgenden Unterkapiteln näher erläutert. Manche Terminplaner haben noch weitere spezielle Ansichtsformen, auf die im abschließenden Unterkapitel „Weitere Darstellungsformen" näher eingegangen wird . Zunächst sollte aber geklärt werden, in welcher Benutzungsoberfläche diese Darstellungsformen zum Einsatz kommen. Da die meisten Oberflächen sehr ähnlich sind, nehmen wir an dieser Stelle als Beispiel Microsoft Outlook. Oben ist eine Symbolleiste zu sehen, über die man die Darstellungsform der Termine schnell ändern kann. Hierüber können auch neue Termine und To-Do's eingefügt werden. (1) Über die Menüleiste sind neben diesen Funktionen noch weitere Funktionen wie Im- und Export von Daten, Hilfe und Ähnliches erreichbar. Rechts oben (2) stellt Outlook zwei Monatskalender zur Verfügung um schnell zwischen den Tagen, die in der Nähe des aktuellen Tages liegen, umschalten zu können. Darunter findet man eine Liste der noch offenen Aufgaben (3). Da Outlook neben der Terminplanerfunktion noch eine E-Mail und Adressbuchfunktion hat, kann man links (4) schnell zu diesen Funktionen wechseln. Da nicht jede Anwendung diese zusätzliche Funktionalität bietet, ist dieser Teil der Oberfläche nur Outlook spezifisch und somit nicht typisch für einen Terminplaner. Im Center des Fensters (5) befindet sich dann eine Ansichtsform für die Termine. Im oben gezeigten Screenshot ist gerade die Tagesansicht ausgewählt worden.

In den nachfolgenden Abschnitten sollen die verschiedenen Darstellungsformen beschrieben werden, die man in den Terminplanern findet. Daher sind die Abbildungen nicht die komplette Benutzungsoberfläche, sondern nur jeweils der Bereich, in dem diese Ansichtsform zu finden ist. (5) [OUT05]

### 2.3.1 Tagesansicht

Die Tagesansicht ist die detaillierteste Form der Termindarstellung. Die Darstellung beschränkt sich nur auf einen Tag. Auf der linken Seite sind die Uhrzeiten eingezeichnet und eingeteilt in 30 Minuten Abschnitte. Wenn ein Termin statt findet wird rechts neben der Uhrzeit von Beginn bis Ende des Termins eine Box gezeichnet. In der Box steht meist der Titel und die genauen Zeitangaben des Termins. In unserem Beispiel wollen wir von zehn bis zwölf Uhr zum Mittagessen in die Mensa gehen. Also wird in diesem Zeitraum eine Box mit dem entsprechenden Inhalt gezeichnet. Wenn mehrere Termine parallel laufen, werden die Boxen in halber Breite nebeneinander dargestellt. Bei Prallelterminen ist die Boxbreite proportional zur Anzahl der Termine. Wenn man mit dem Cursor über den Termin geht, erhält man meist noch weitere Informationen über den Termin, wie z.B. die ersten 100 Zeichen der Beschreibung. In unserem Beispiel ist nur in der Beschreibung vermerkt, dass wir in die Mensa gehen wollen. Wenn man mit der Maus über die Box geht, wird diese Information angezeigt. An Werktagen ist die Arbeitszeit in einer anderen Farbe gekennzeichnet, als die übrige Zeit. In unserem Beispiel liegt die Arbeitszeit zwischen 8 und 16 Uhr. Der Benutzer kann in den Optionen der Systeme seine individuelle Arbeitszeit einstellen. Wenn man zum Beispiel um 16.30 einen neuen Termin eintragen möchte, klickt man an dieser Stelle mit der Maus einfach rechts und kann „Neuer Termin" auswählen. Dies erspart einem, den Beginn des Termins manuell einzugeben. Dieses Feature ist in vielen Kalendersystemen vorhanden. [KOG05]

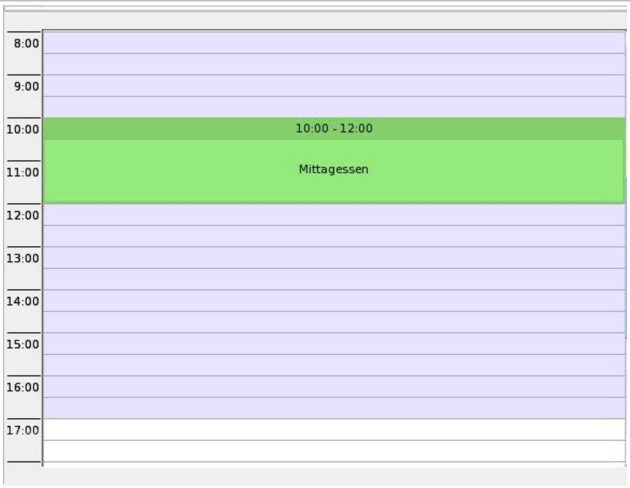


*Abb. 9 - Ausschnitt aus der Tagesansicht in KOrganizer*

### 2.3.2 Wochenansicht

Wem die tägliche Übersicht beim Planen der kommenden Woche nicht ausreicht, dem bietet sich die Möglichkeit der Wochenansicht. Hier hat man den Überblick über eine ganze Woche. Diese Ansicht ist typischerweise tabellarisch angeordnet. Auf der vertikalen Achse sind die Uhrzeiten abgetragen und auf der horizontalen Achse die Wochentage. Wenn man hier mit der Maus über die Termine geht, erhält man nähere Informationen zum Termin.

In dieser Ansicht ist es nicht mehr möglich, jeden Termin anhand seines Beginns und seines Endes genau einzuzeichnen. Auch werden parallele Termine in dieser Ansicht nicht mehr nebeneinander dargestellt. Daher erhält man im Informationsfenster, das erscheint, wenn man mit der Maus über den Termin geht, seinen genauen Beginn und sein genaues Ende, sowie dessen Titel.

Wie wir in unserem Beispiel sehen, wird der heutige Tag in der Woche in einer anderen Farbe dargestellt. Im webbasierten Kalender von web.de wird der heutige Tag in rot dargestellt. Diese Darstellungsform ist eher untypisch. In vielen anderen Systemen wird das komplette Wochenende in einer anderen Farbe dargestellt, als die übrigen Tage.

Zusätzlich werden Sonntage und Feiertage in Rot gekennzeichnet. Dem aktuellen Tag wird dann meist die Farbe blau zugewiesen. In den meisten Fällen hat der Benutzer in den Optionen der Kalendersysteme die Möglichkeit, die Farbvergabe für die einzelnen Tage selbst festzulegen.

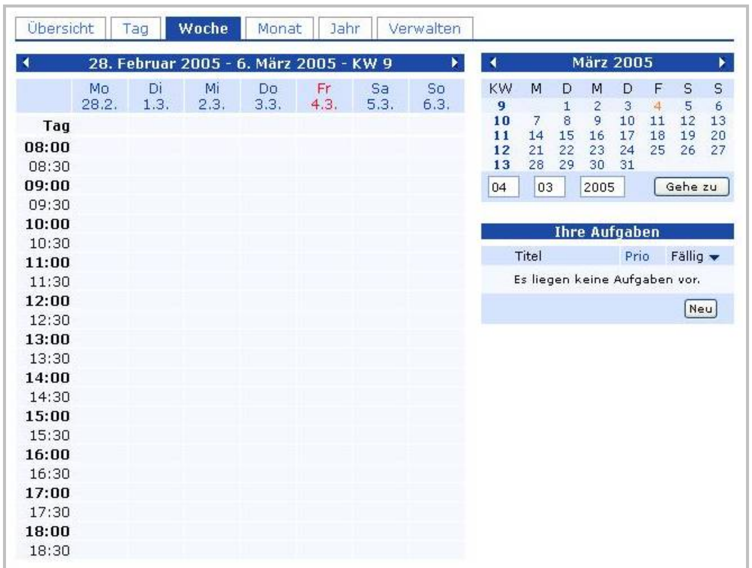


*Abb. 10 - Wochenansicht im webbasierten Kalender von Web.de*

Wenn man viele Termine hat kann diese Übersicht unter Umständen unübersichtlich werden. Dann kann man jedoch wieder auf die zuvor beschriebene Tagesansicht zurückgreifen, wenn man sich die Termine eines Tages näher betrachten will. Um in die Tagesansicht zu wechseln, kann man in manchen Systemen einfach den entsprechenden Tag in der Wochenansicht oben anklicken. [DEWEB]

### 2.3.3 Monatsansicht

Wer dies immer noch nicht übersichtlich genug findet, hat als dritte Variante noch die Monatsansicht zur Verfügung. Hier hat man den Überblick über alle Termine des Monats. Leider steht in dieser Ansicht für jeden Tag nur sehr wenig Platz zur Verfügung. An jedem Tag werden die Termine nacheinander, sortiert nach ihrem Beginn, dargestellt. Wenn an den Tagen sehr viele Termine anliegen, kann es sein, dass nicht alle Termine angezeigt werden, oder jeder Termin wird nur noch durch ein kleines Symbol gekennzeichnet. Diese Ansicht gibt nur einen groben Überblick. Wenn man Termine hat, die über mehrere Tage, oder gar Wochen gehen, kann man deren Beginn und Ende in dieser Ansicht am Besten verfolgen. Diese Ansicht eignet sich zum Beispiel zum Planen des Urlaubs oder ähnlichen längerfristigen Terminen. Hat man jeden Tag viele kürzere Termine, empfiehlt es sich auf eine andere Darstellungsform , wie Tagesansicht oder Wochenansicht, zurückzugreifen.

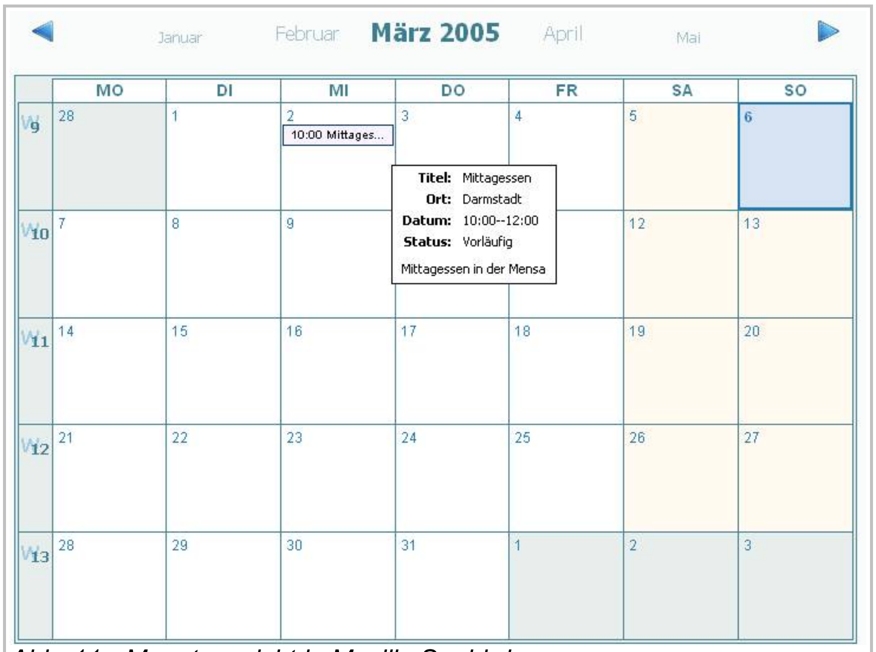


*Abb. 11 - Monatsansicht in Mozilla Sunbird*

Auf der vorigen Seite ist die Monatsansicht in Mozilla Sunbird dargestellt. Hier sehen wir auch die typische Farbvergabe für die einzelnen Tage, die bereits in Kapitel 2.3.2 diskutiert wurde. Die Wochenenden haben eine andere Farbe wie die restlichen Tage der Wochen. Die Abbildung wurde am Sonntag, den 6. März angefertigt. Daher ist dieser Tag auch blau eingefärbt. Unser Mittagessenstermin ist auch am Mittwoch in der ersten Märzwoche zu sehen. Wenn man mit der Maus darüber geht, erhält man in Form eines Tooltipps weitere Informationen. Dies wurde in den anderen Ansichtsformen auch schon angesprochen, in dieser Abbildung ist zu erkennen, wie solch ein Tooltipp aussehen kann.[SUN02]

## 2.3.4 Weitere Darstellungsformen

Neben diesen Darstellungsformen bieten die verschiedenen Kalendersysteme meistens noch weitere Möglichkeiten die Termine auf dem Bildschirm auszugeben. Unten sind anhand zweier Beispiele weitere Ansichten gegeben. Dieses Kapitel kann nur einen kleinen Einblick gegeben, welche weiteren Ansichten möglich sind.

Web.de bietet seinen Kunden zum Beispiel auch noch eine Jahresansicht an. In dieser Ansicht können jedoch keine Termine mehr dargestellt werden. Es ist nur noch möglich Tage mit Terminen in einer anderen Farbe darzustellen, wie terminfreie Tage. In dieser Ansicht hat man zum Beispiel einen guten Überblick über die Feiertage des Jahres.

Microsoft Outlook beinhaltet zusätzlich eine spezielle Art der Wochenansicht: Hier werden nur die Tage der Arbeitswoche angezeigt. Also ohne das Wochenende. (siehe unten)

In Mozilla Sunbird hingegen findet man eine Multi-Wochenansicht in der man sich die Termine der kommenden 2 Wochen anschauen kann. Wie viele Wochen in dieser Ansicht dargestellt werden, kann in den Optionen festgelegt werden.

Wie man sieht gibt es viele Arten, die Termine zu visualisieren, aber alle weiteren Arten der Ansichten sind mehr oder weniger von den drei Grundansichten Tagesansicht, Wochenansicht und Jahresansicht abgeleitet. [SUN02] [DEWEB] [OUT05]

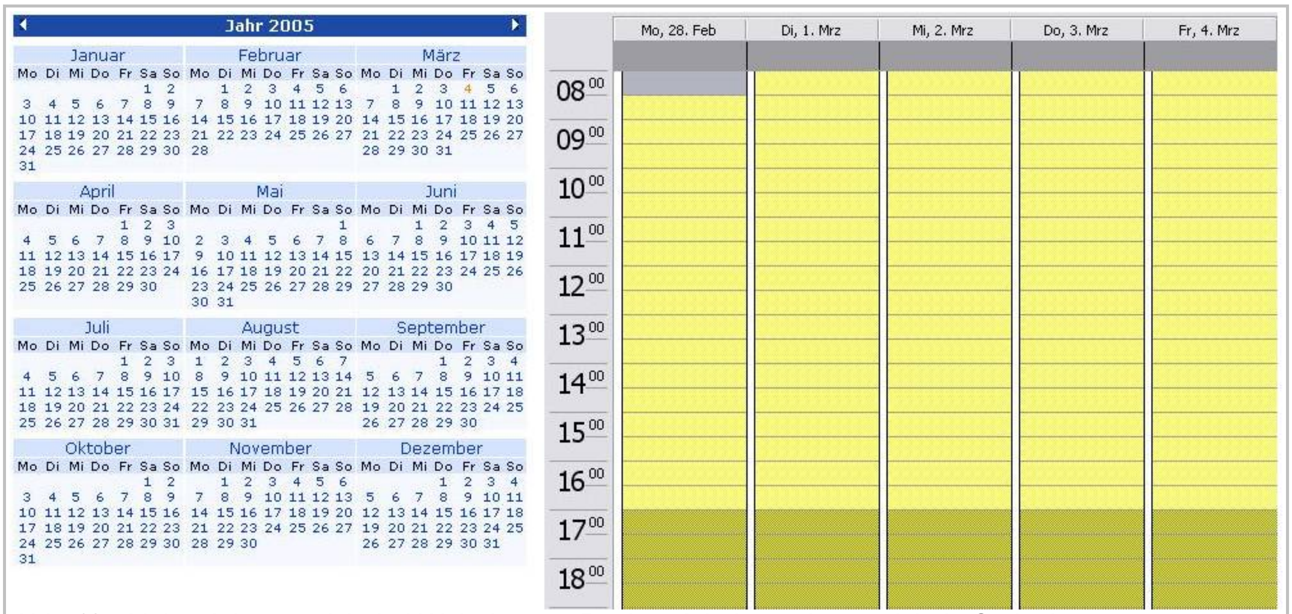



*Abb. 12 - Links: Jahresansicht bei Web.de Rechts: Arbeitswochenansicht in Outlook*

## 2.4 Im- und Export von Daten

In diesem Abschnitt soll etwas näher darauf eingegangen werden, auf welche Arten die Kalendersysteme ihre Daten Im- und Exportieren. In erster Linie hat diese Funktion den Zweck der Datensicherung. Gerade wenn man viele Termindaten gespeichert hat, will man hin und wieder seine Daten auf externen Datenträgern sichern. Die verschiedenen Dateiformate, in die man seine Kalenderdaten auslagern kann, sind sehr unterschiedlich. Daher wollen wir uns an dieser Stelle auf die drei häufigsten Dateiformate beschränken. Als kleines Beispiel ist in Abbildung 13 der Ex- und Importassistent von Microsoft Outlook dargestellt. Da Outlook neben einem Terminplaner auch noch ein E-Mail Verwaltungssystem und ein Adressbuch beinhaltet, hat man hier mehr Möglichkeiten des Im- und Exports von Daten, die mit Terminen jedoch nichts zu tun haben. Daher wird an dieser Stelle nicht näher auf diese Möglichkeiten eingegangen. [OUT05]




*Abb. 13 - Im- und Exportassistent in Outlook*

Sehr oft hat man die Möglichkeit, seine Daten in ein so genanntes CSV Format auszulagern. CSV steht für Character Separated Values oder auch Comma Separated Values. Damit ist eine Textdatei gemeint, in der die Datensätze tabellarisch abgelegt sind. Meistens sind die einzelnen Werte durch Kommas getrennt. Jedoch ist es im Prinzip möglich, jedes beliebige Zeichen als Trenner zu verwenden. Kommt dieses Trennzeichen in einem Datenelement vor, muss es zuvor maskiert werden. Meistens stehen in der ersten Zeile die Namen der verschiedenen Attribute und in den folgenden Zeilen dann die Daten. Dabei wird jeder Zeichensatz durch einen Zeilenumbruch getrennt. Einen offiziellen Standard gibt es für dieses Dateiformat leider nicht. Unten rechts ist ein kleines Beispiel gegeben, wie Termindaten in solch einer CSV Datei aussehen könnten. Das Beispiel ist vereinfacht. Termine haben normalerweise wesentlich mehr Attribute. Die Zeiten sind im Format YYYY-MM-DD HH:MM gespeichert. Da es keinen festen Standard gibt und viele Anwendungen ihre Attribute unterschiedlich benennen und auch in unterschiedlicher Reihenfolge, sind die Dateien meist inkompatibel zueinander. Das heißt, wenn man in Terminplaner A seine Daten ins CSV Format exportiert, kann Terminplaner B diese Daten nicht einlesen, obwohl er auch das CSV Format unterstützt. Da Microsoft Outlook eine sehr verbreitete Anwendung zur Terminplanung ist, verstehen jedoch viele Anwendungen die Notation der mit Outlook ausgelagerten CSV-Dateien. [WIKICSV]

```
Beginn, Ende, Titel
200505031000, 200505031200, Mittagessen
200506031400, 200506031800, Vorlesung
200506031800, 200506031930, Nachhilfe
```

Als zweites Format findet man meist das ICS Format. iCalendar ist ein Standard zum Austausch von Kalenderinformationen und wird im RFC 2445 definiert. Da in Kapitel fünf noch ausführlich die Synchronisation von Kalenderdaten diskutiert wird und hierbei auch das ICS-Format angesprochen wird, wird an dieser Stelle nicht näher darauf eingegangen. An dieser Stelle sei nur gesagt, dass Anwendung A die exportierten Kalenderdaten ins ICS Format von Anwendung B lesen kann.

An dritter Stelle ist dann noch das programminterne Format zu nennen in dem die Anwendung die Termindaten speichert. Das Format ist von Anwendung zu Anwendung verschieden. Mozilla Sunbird speichert zum Beispiel auch intern im ICS Format. [SUN02]

## 2.5 Weitere Komponenten

Nachdem die wesentlichen Komponenten eines Kalendersystems aufgelistet wurden, soll jetzt noch einmal kurz darauf eingegangen werden, welche Funktionalitäten es gibt, die nicht jedes Kalendersystem beinhaltet. Da es eine Fülle von solchen Funktionen gibt, kann dies hier nur an Stelle von ein paar Beispielen erfolgen.

Bereits in der Einleitung dieser Bachelor Arbeit wurde erwähnt, dass es Kalendersysteme gibt, bei denen man einen Termin auch an alle Beteiligten per E-Mail schicken kann und dieser Termin dann beim Empfänger automatisch eingetragen wird. Diesen Mechanismus nennt man Group Scheduling. Da Outlook neben dem Terminplaner auch E-Mails unterstützt, kann man mit Outlook Termine per Group Scheduling erstellen. Wenn der Empfänger der E-Mail ebenfalls Outlook verwendet oder eine andere Anwendung, die diese Art der Terminplanung versteht, wird der Termin beim Empfänger automatisch in den Kalender eingetragen. Als Transfersprache von Group Scheduling Terminen kommt meist auch das ICS Format zum Einsatz, das im vorigen Kapitel erwähnt wurde und in Kapitel fünf ausführlich behandelt wird. In Outlook kann man an die Termine Dateien anhängen, die beim Group Scheduling dann auch per Mail weitergeleitet werden. Eine weitere Anwendung, die Group Scheduling unterstützt, findet man zum Beispiel unter Suse Linux (KOrganizer).

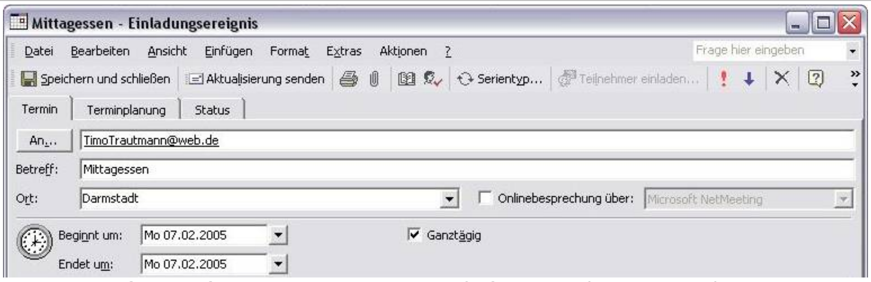


*Abb. 14 - Group Scheduling in Microsoft Outlook (Ausschnitt)*

Beim Erstellen von Terminen stellt Outlook dem Benutzer einen Planungsassistenten zur Verfügung, der dabei hilft einen passenden Zeitpunkt für den Termin zu finden. [OUT05]

Als letztes Beispiel schauen wir uns den webbasierten Terminkalender von Freenet an. Viele Benutzer tragen Geburtstage auch in ihren Terminplaner ein. Hier wird eine spezielle Ansicht geboten, in der nur Termine dargestellt werden, die sich in der Kategorie Geburtstag befinden. Der aktuelle Monat wird dabei in einer anderen Farbe dargestellt, als die übrigen Monate. Leider sind in unserem Terminplaner in Abbildung 15 keine Geburtstage gespeichert. Diese Art der Ansicht ist auch noch in anderen Terminplanern zu finden und wird meist als Geburtstagskalender bezeichnet. [FREESCH]

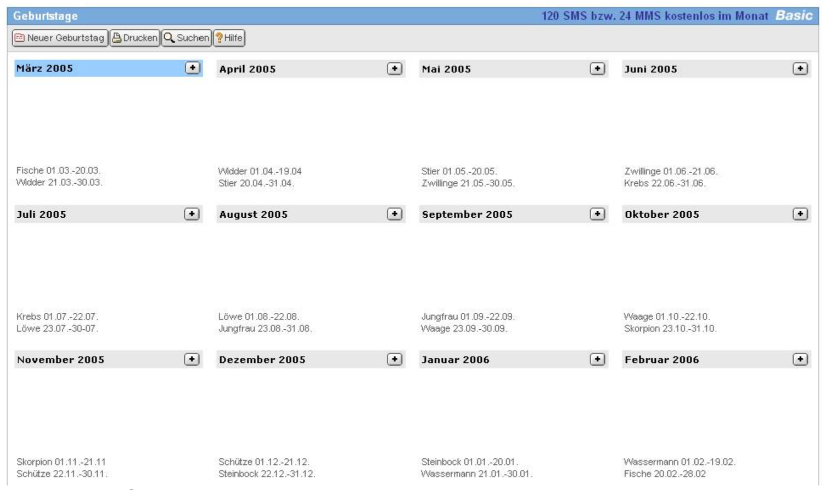


*Abb. 15 - Geburtstagskalender von Freenet*

Wie man sieht gibt es viele Erweiterungen und Sonderfunktionen, die jedem Terminplaner etwas Einzigartiges geben, obwohl sie alle die Basisfunktionalitäten eines Kalendersystems unterstützen.

# 3 Einsatz von Kalendersystemen in der Gesellschaft

Nachdem wir im letzten Kapitel bereits existierende Kalendersysteme und deren Komponenten kennen gelernt haben, soll in diesem Kapitel der Benutzer in den Vordergrund gerückt werden. Es soll gezeigt werden, welche Anforderungen der Benutzer an das System hat, wann er diese Systeme einsetzt und worin Vorteile bei der Nutzung dieser Systeme bestehen.

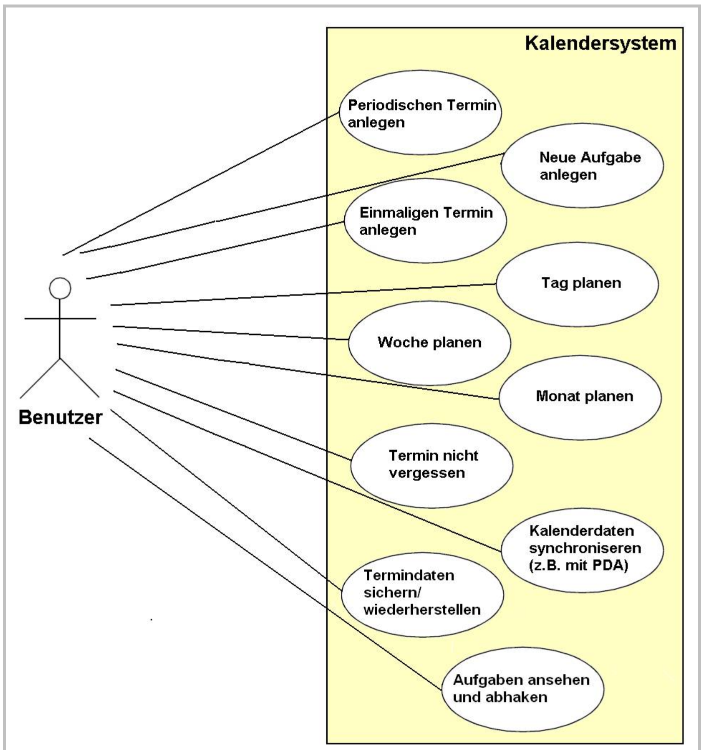


*Abb. 16 - Anforderungen an Kalendersysteme*

Zunächst wollen wir die Anforderungen des Benutzers den Systemkomponenten zuordnen. Das rechts stehende Use-Case Diagramm zeigt die wesentlichen Anforderungen des Benutzers an das System. Der Benutzer möchte neue Termine anlegen, die einmal oder auch öfters statt finden. Das System stellt hierzu eine Eingabemöglichkeit für einmalige- und periodische Termine zur Verfügung. Manchmal hat der Benutzer eine Aufgabe zu erledigen und möchte diese auch gerne vermerken, damit er sie nicht vergisst. Hierzu kann man in vielen Kalendersystemen neben Terminen auch To-Do's anlegen. Und damit der Benutzer diese Termine und Aufgaben nicht verpasst, möchte er zusätzlich vor jedem Termin noch einmal daran erinnert werden. Dazu stellt das System bei der Eingabe auch die Möglichkeit der Terminerinnerung zur Verfügung. Oft hat er dann den Wunsch nicht nur Überblick über seine Termine des aktuellen Tages zu bekommen, sondern auch einen Überblick über die Woche oder sogar den ganzen Monat. Hierzu stellen die Kalendersysteme die in Kapitel zwei beschriebenen Ansichten Tages-, Wochen- und Monatsansicht bereit. Natürlich möchte er auch einen Überblick über seine Aufgaben und möchte diese bei Bedarf als erledigt kennzeichnen. Schließlich möchte er noch seine Termindaten mit seinem Handy oder PDA synchronisieren, damit er die Termine auch unterwegs zur Hand hat. Hierzu gibt es in den Systemen in der Regel eine Import- und Exportmöglichkeit von Termindaten. Diese Im- und Export Komponente erfüllt zusätzlich den Zweck der Datensicherung. Manche Anwendungen beinhalten sogar eine direkte Schnittstelle zwischen PDA und Kalendersystem. Dies sind die wesentlichen Anforderungen des Benutzers an Kalendersysteme. In Kapitel 2.5 wurde auch die Möglichkeit diskutiert, Termine per E-Mail zu versenden und sie beim Empfänger automatisch in den Terminplaner einzutragen. Da diese Anforderung nicht in allen Kalendersystemen erfüllt ist, wurde dieser Use-Case weg gelassen.

In den folgenden Kapiteln soll zunächst dargestellt werden, in welchen Situationen Benutzer mit Kalendersystemen umgehen, um hieraus darauf zu schließen, auf welchen Systemumgebungen solche Systeme sinnvoll sind. Danach soll noch einmal konkret geklärt werden, welche Vorteile diese Systeme gegenüber dem herkömmlichen Terminbuch haben. Dieser Abschnitt wird abschließend durch eine Fallstudie vertieft.

## 3.1 Typische Situationen für den Einsatz von Kalendersystemen

Dieser Abschnitt soll etwas näher darstellen, in welcher Situation Menschen ein Kalendersystem nutzen und welche der in Kapitel zwei beschriebenen Komponenten in dieser Situation zum Einsatz kommen. Da es praktisch unendlich viele Situationen gibt, in denen man ein Kalendersystem verwenden kann, beschränkt sich dieser Abschnitt nur auf die wesentlichen und häufigsten Situationen.

Um uns ein paar typische Situationen für den Einsatz von Terminplanern vorstellen zu können, denken wir uns eine fiktive Situation aus. Es gibt einen Service Techniker, der im Team vor Ort Serviceleistungen für einen großen PC Hersteller tätigt. Meistens handelt es sich dabei um defekte Desktop PCs. Begleiten wir doch einmal diesen Service Techniker durch einen Arbeitstag. Damit er seine Termine bei den Kunden besser im Auge behalten kann, besitzt er ein PDA. Gleich beim ersten Kunden gibt es ein Problem, da er das beschädigte Teil nicht dabei hat. Er muss den Kunden vertrösten und macht mit ihm ein Treffen in der nächsten Woche aus. Er trägt diesen Termin in sein PDA ein und schaut nach dem nächsten Termin. Auf dem Weg zum nächsten Kunden wird er von seinem Chef angerufen, dass es noch einen dringenden Fall gibt. Da die Wohnung des Kunden in der Nähe liegt und er auch Nachmittags noch etwas Luft hat, stimmt er zu. Auch diesen Termin trägt er in seinem PDA ein. Beim nächsten Kunden läuft alles planmäßig und das Problem ist schnell behoben. Zwischenzeitlich wird er von einem Kunden angerufen, den er eigentlich um 14.00 treffen sollte. Der Kunde erklärt, dass er kurzfristig eine persönliche Angelegenheit zu erledigen hätte und könne den Termin daher nicht wahrnehmen. Der Service Techniker gibt dem Kunden einen neuen Termin am darauf folgenden Tag und hält die Terminverschiebung in seinem PDA fest. Auf dem Weg zum letzten Kunden für diesen Tag ruft ihn seine Frau an und bittet ihn eine Bank-Überweisung für sie zu tätigen und gibt ihm die Daten für die Überweisung. Des Weiteren wäre sie am Abend bei einer Freundin und er solle seinen Sohn bei einem Freund um 19.00 Uhr abholen. Er willigt ein, gibt die Abholzeit in Form eines Termins und die Überweisung in Form eines To-Do's in sein PDA ein. Nach dem letzten Kunden hat er mit seinen Kollegen noch ein Meeting beim Teamleiter. Am Ende der Besprechung legen sie noch den Termin für das nächste Teammeeting und für ein Geschäftsessen fest. Auch diese Termine hält der Service Techniker auf seinem PDA fest. Nachdem er seinen Sohn abgeholt hat, möchte er einen Überblick über die Termine der kommenden Tage gewinnen. Da das Display des PDAs sehr klein ist, synchronisiert er seine Termindaten mit seinem Einzelplatzrechner, auf dem er ein Kalendersystem installiert hat. Jetzt hat er eine detaillierte Tages- und Wochenansicht und plant die Route, die er am folgenden Tag fahren möchte. Der Service Techniker ist Mitglied in einem Fußball Verein. Er surft auf die Seite des Vereins und lädt sich die aktuellen Termine für Sitzungen und Training herunter und importiert diese in seinen Kalender auf dem Einzelplatzrechner. Danach trägt er provisorisch in seinen Terminplaner auf dem PC einen möglichen Urlaub ein. Da er über Online-Banking verfügt, nimmt er jetzt die Überweisung für seine Frau am PC vor und markiert danach die Aufgabe als erledigt. Um wieder aktuelle Termindaten auf dem PDA zu haben, synchronisiert er seine Termindaten zwischen PC und PDA ein zweites Mal, bevor ein arbeitsreicher Tag zu Ende geht.

Dieses Beispiel ist nicht nur typisch für einen Service Techniker, sondern für alle Leute, die in ihrem Beruf sehr mobil sind und bei denen kein Tagesablauf dem anderen gleicht. Natürlich kann man sich dann nicht mehr alle Termine merken und möchte diese immer schnell zur Hand haben. Dass dies bei einem Arbeitnehmer, dessen täglicher Arbeitsablauf in der Regel immer gleich ist, nicht so wichtig ist, sollte auch klar sein.

Im Wesentlichen haben wir in unserem Beispiel zwei Situationen heraus stellen können, in denen Menschen mit Terminplanern umgehen: Während sie unterwegs sind und während sie im Büro oder Zuhause sitzen.

Unterwegs ist es meist nicht wichtig einen Überblick über die nächsten Tage zu haben. Viel wichtiger ist es, zu sehen, welche Termine als nächstes anliegen. Auch ist es wichtig eine Möglichkeit zu haben, neue Termine oder Änderungen festzuhalten. Unterwegs bleibt meist wenig Zeit und man möchte immer seine Termine zur Hand haben. Daher eignet sich ein Desktop PC weniger. Sinnvoller ist hier ein PDA, das schnell eingeschaltet ist und in jede Hosentasche passt.

Wenn man im Gegensatz dazu etwas mehr Ruhe hat, also zum Beispiel Zuhause oder im Büro, möchte man auch die folgenden Tage planen, oder Termine per E-Mail abklären und eintragen. Oder man möchte weitere Termine aus dem Internet besorgen. Hierzu eignet sich das kleine Display des PDAs weniger. Daher möchte man seine Termine auch auf den PC übertragen können und wieder zurück. So hat man neben einem großen Bildschirm auch einen leistungsfähigen Rechner zur Verfügung, um eine bessere Terminübersicht zu haben.

An dieser Stelle zeichnet sich schon ab, dass Terminplaner auf verschiedenen Plattformen meist nicht alleine genutzt werden, sondern noch auf einer zweiten Plattform. Zum Beispiel eben lokaler Einzelplatzrechner und PDA. Erst im Team werden diese Kalendersysteme zu einem starken Helfer für den Menschen. Auch ist hieraus zu erkennen, dass Synchronisation zwischen diesen Systemen sehr wichtig ist. Daher werden genau diese zwei Themen, Plattformen und Synchronisation in den zwei folgenden Kapiteln behandelt.

## 3.2 Vor- und Nachteile zwischen Kalendersystemen und Terminbüchern

Bereits in der Einführung wurde auf das klassische Terminbuch hingewiesen. An dieser Stelle soll jetzt etwas näher darauf eingegangen werden, ob die neuen Kalendersysteme immer ein Segen sind, oder ob es zu den herkömmlichen Terminbüchern noch enorme Nachteile gibt.

Auf den ersten Blick ist es kaum von der Hand zu weisen, dass es komplizierter ist, einen Termin in ein Kalendersystem einzutragen, als in ein Terminbuch. Wenn sich das Kalendersystem auf einem Einzelplatzrechner befindet, muss dieser hierzu ggf. erst gebootet werden, was viel Zeit in Anspruch nimmt. Bei Kalendersystemen auf dem Handy oder dem PDA ist es meist sehr unkomfortabel Buchstaben einzugeben, so dass dies auch jede Menge Zeit in Anspruch nimmt. Das Terminbuch ist schnell aufgeschlagen und der Termin ist an der entsprechenden Stelle eingetragen.

Im Terminbuch kann man schnell zwischen dem ersten und letzten Monat des Jahres umblättern. In seinem technischen Bruder sind hierzu unter Umständen mehrere Klicks oder eine Eingabe notwendig.

Stichwort Verfügbarkeit: Im folgenden Kapitel sollen die Vor- und Nachteile der verschiedenen Plattformen, auf denen Kalendersysteme zu finden sind, näher erörtert werden. Daher soll dieses Thema hier nur kurz angeschnitten werden. Natürlich kann man seinen Terminplaner auf dem lokalen Einzelplatzrechner nicht mal eben in die Hosentasche stecken und mitnehmen. Auch ist es nicht in jeder Situation möglich, seinen Laptop auszupacken und einzuschalten. (Zum Beispiel auf Reisen). In diesem Punkt schaffen die kleinen Handcomputer (PDAs) oder Handys zwar Abhilfe, aber jeder kennt

bestimmt eine Situation in der der Akku des Handys leer war und er daher nicht mehr telefonieren konnte. Eine sehr ärgerliche Situation. Ähnlich ist es auch bei den Kalendersystemen. Ist der Akku leer und besteht keine Möglichkeit, diesen aufzuladen oder gegen einen vollen auszutauschen, hat man keinen Zugriff mehr auf seine Termine. Das Terminbuch hingegen ist immer verfügbar und normalerweise auch nicht so groß, dass man es nicht mitnehmen könnte. Natürlich kann man, um dem entgegen zu wirken, zusätzliche Ausdrucke seiner Termine mitnehmen, jedoch ist dies mit weiterem Aufwand verbunden. Wenn man regelmäßig solche Ausdrucke anfertigt ist es unter Umständen dann wieder praktischer, gleich auf das Terminbuch zurück zu greifen.

Bei all diesen Nachteilen fragt man sich jedoch, warum es denn dann überhaupt Kalendersysteme gibt und warum Diese so beliebt sind. In der Tat gibt es auch eine Reihe von Vorteilen, die jetzt aufgelistet werden sollen:

Dass es aufwendiger ist einmalige Termine einzutragen, wurde bereits diskutiert. Was ist aber mit periodischen Terminen? Stellen wir uns vor wir müssten unseren wöchentlichen Saunatermin jede Woche ins Terminbuch eintragen. Nicht vorzustellen. Im technischen Bruder gibt man den Termin ein mit dem Vermerk wöchentlich und er wird automatisch wiederholt. Hier geht das Anlegen eines Termins eindeutig schneller, als im Terminbuch.

Was ist, wenn wir die kommende Woche mit dem Terminbuch planen wollen. Oft müssen wir umblättern, um einen Überblick zu erhalten. In Terminbüchern sind meist 1-2 Seiten für einen Tag reserviert. Dies ist äquivalent zur Tagesansicht in Kalendersystemen. Wie wir in Kapitel 2 gesehen haben, hat man neben der Tagesansicht noch eine Wochen- und Monatsansicht, die es erleichtern, die Wochen und Monate zu planen.

Es könnte sein, dass man rückblickend noch einmal seine Termine des vergangenen Jahres ansehen möchte. Ein Terminbuch geht in der Regel nur über ein Jahr und man hat auch das Terminbuch des letzten Jahres nicht zur Hand. Da Termindaten nicht sehr viel Speicherplatz in Anspruch nehmen (Ausnahme: Jedem Termin sind umfangreiche Dateianhänge hinzugefügt) ist es kein Problem, die Termine der vergangenen Jahre im Kalendersystem zu belassen. So kann man auf Diese jederzeit zurückgreifen. Auch wenn es sehr unwahrscheinlich ist, könnte der Benutzer das Bedürfnis haben, einen Termin einzutragen, der erst mehrere Jahre in der Zukunft statt findet. Da Terminbücher meistens nur das aktuelle Jahr beinhalten, ist dies nicht ohne weiteres möglich. In Kalendersystemen ist es mit wenigen Klicks möglich, zehn Jahre in die Zukunft zu springen. Bei Bedarf auch mehr.

Auch wurde die Situation einer Terminverlegung noch nicht angesprochen. Im Terminbuch wird der Termin meist weggestrichen und an anderer Stelle wieder eingetragen. Geschieht dies öfters kann es sehr unübersichtlich werden. In der digitalen Version kann man Termine einfach bearbeiten: Wenn man den Terminen einen neuen zeitlichen Rahmen zuweist, werden sie in der Ansicht einfach verschoben. Wenn man sie löscht verschwinden sie einfach. Dieser Vorteil wird besonders spürbar, wenn man über drei Monate einen täglichen Termin in sein Terminbuch eingetragen hat, der dann doch in diesen drei Monaten nicht statt findet.

Und was ist mit Begleitinformationen zu Terminen, wie zum Beispiel einer Beschreibung? In Terminbüchern steht hierzu nur begrenzt Platz zur Verfügung. In Kalendersystemen hingegen ist die Länge einer solchen Beschreibung nahezu unbegrenzt. Auch ist es hier manchmal möglich Dokumente in digitaler Form an den Termin anzuhängen.

Und schließlich gibt es da noch die Sonderfunktionen, auf die schon in Kapitel zwei hingewiesen wurde. Man kann einen Termin anlegen und die Beteiligten per E-Mail

darüber informieren. Dieser Termin wird dann vollautomatisch in den Terminplaner des Empfängers eingetragen.

Immer sind wir davon ausgegangen, dass wir unsere Termine selbst in den Terminplaner eingeben. Bei einem Terminbuch ist das in der Tat die einzige Möglichkeit. Wie sieht es aber mit den digitalen Systemen aus? Denkbar ist zum Beispiel, dass ein Theater seine Aufführungen und deren zeitlichen Rahmen als Datei zur Verfügung stellt und man diese Termine einfach in sein Kalendersystem importiert und fertig. Oder die Firmensitzungen und Besprechungen werden den Mitarbeitern einfach als Datei zum importieren in ihren Kalender zur Verfügung gestellt, ohne dass man sie selbst eintragen muss. Ein weiteres Anwendungsbeispiel sind Vereine, die die nächsten Versammlungen und Trainingszeiten ihren Mitgliedern in digitaler Form zur Verfügung stellen. Als Dateiformat kommt hier das ICS Format zum Einsatz, das in Kapitel fünf ausführlich behandelt wird. An dieser Stelle seien ein paar konkrete Beispiele genannt:

Der Musikverein Durlach-Aue besitzt unter http://www.mv-aue.de/index2.html eine Internetpräsenz. Unter Termine kann der Besucher Einsicht in die kommenden Termine nehmen und hat auch die Möglichkeit, sich die Termine als ICS Datei herunter zu laden. [DUAU]

Die Musikband „Herr Press“ ist unter http://www.elmstreet.de erreichbar. Fans können dort die nächsten Auftrittstermine der Band für ihren Terminplaner herunter laden. [HRPRESS]

Im Internet gibt es eine OpenSource Applikation namens TV-Browser. Diese Applikation ermöglicht es eine aktuelle TV-Zeitschrift auf den PC zu laden und anzuschauen. Da diese Software frei verfügbar ist, gibt es auch eine Anzahl von Plugins dafür. Ein Plugin exportiert eine ausgewählte Sendung ins ICS Format. Somit kann man auch seine TV Sendungen, ohne größeren Aufwand, in seinen eigenen Kalender übernehmen. [TVBROW]

An der Fachhochschule Darmstadt verwenden wir zum belegen der Lehrveranstaltungen ein Onlinebelegsystem. Hier können sich die Studenten auch die Termine für Vorlesungen, Praktika und Klausuren im ICS Format für ihren Terminplaner herunterladen. Als Abschluss dieses Abschnittes soll dieses Onlinebelegsystem in Form einer Fallstudie näher erklärt werden. [FHDOBS]

### 3.2.1 Fallstudie: Das OBS der Fachhochschule Darmstadt

Jeder Student an der Fachhochschule Darmstadt muss im Fachbereich Informatik zunächst seine Lehrveranstaltungen belegen, bevor er diese besuchen darf. Früher wurde dies über Papier Belegscheine gemacht. Heute gibt es an der Fachhochschule ein Onlinebelegsystem, kurz OBS. Jeder Student gibt während einer festgelegten Zeitspanne seine Belegwünsche über das Internet ein. Also an welchen Lehrveranstaltungen er teilnehmen möchte. Nach Anmeldeschluss vergibt das System die freien Plätze dann an die Bewerber dieser Lehrveranstaltung. Des öfteren gibt es mehr Bewerber, als freie Plätze. Daher werden die Plätze mit Hilfe eines Zuteilungsalgorithmus vergeben. Ein Student, der sich in Regelstudienzeit befindet, hat zum Beispiel Vorrang vor einem Studenten im 15. Semester. Es gibt noch viele weitere Randbedingungen, nach denen die Plätze vergeben werden. Diese Randbedingungen sollen jedoch hier nicht weiter diskutiert werden, da Algorithmen nicht Thema dieser Abschlussarbeit sind. Nach Abschluss dieses Verfahrens weiß der Student, in welchen Praktika und in welchen Vorlesungen er einen

Platz bekommen hat. Er hat die Möglichkeit, das Ergebnis online abzufragen. Nach Verteilung der Plätze haben die Studenten über eine gewisse Zeitspanne die Möglichkeit, Plätze die sie zugeteilt bekommen haben, aber nicht mehr benötigen, frei zu geben. Wiederum andere Studenten, die gerne einen Platz bekommen hätten können diesen dann noch im Nachhinein über das OBS erhalten.

Die Dozenten der Lehrveranstaltungen bekommen dann Teilnehmerlisten für ihre Lehrveranstaltungen. Auch ist es im Fachbereich Informatik mittlerweile Pflicht, dass sich jeder Student für eine Klausur zuvor anmeldet, wenn er mit schreiben möchte. Damit dies nicht so viel Chaos und Papierberge verursacht, findet diese Anmeldung zu den Leistungsnachweisen ebenfalls über das OBS statt. Auch hier erhalten die Dozenten Listen mit den Teilnehmern, damit sie wissen, auf welche Zahl sie sich dann bei der Klausur einstellen müssen.

Früher musste man seinen Stundenplan dann selbst schreiben und die Termine selbst in seinen Terminplaner eintragen, sofern man einen hatte. Heute wird jedem Studenten auch ein persönlicher Stundenplan über das OBS zur Verfügung gestellt, den er nur noch ausdrucken muss.

Neben diesem neuen Feature, kann der Student auch die Termine seiner Vorlesungen, Praktika und Klausuren als Datei im ICS Format herunter laden. Diese kann man dann in vielen Terminplanern einfach importieren und muss sie nicht mehr mühselig manuell eintippen.

Als allerneustes Feature muss man jetzt noch nicht einmal mehr einen Ausdruck seiner Noten im Sekretariat abholen, um seine Noten zu kontrollieren. Neuerdings bietet das OBS sogar direkte Einsicht über alle Noten des jeweiligen Studenten. [FHDOBS]

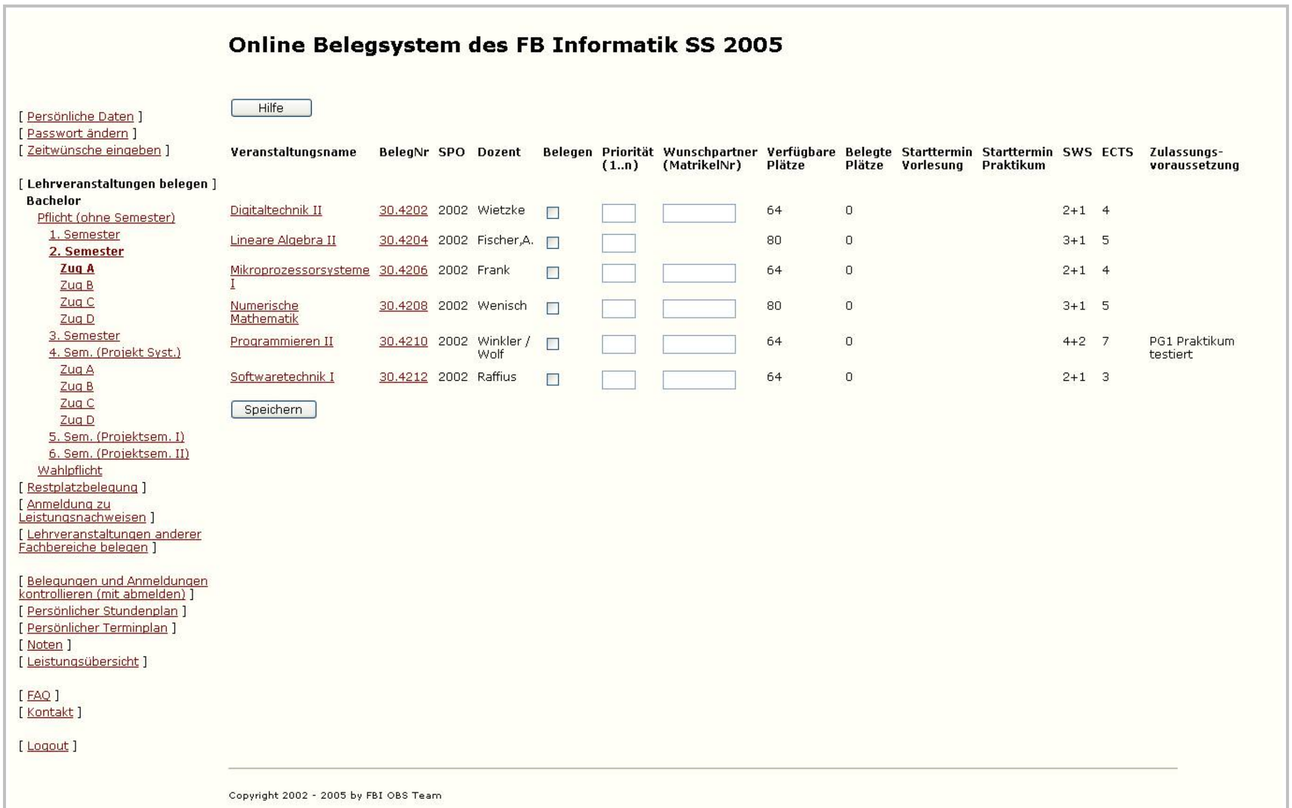



*Abb. 17 - OBS des Fachbereichs Informatik an der Fachhochschule Darmstadt*

# 4 Kalendersysteme auf verschiedenen Plattformen

Im vergangenen Kapitel haben wir viel über Kalendersysteme und deren Einsatz in der Gesellschaft gehört. Wir haben mitbekommen, das man Kalendersysteme nicht nur im Büro oder daheim verwendet, sondern des öfteren auch auf Reisen oder unterwegs. Dabei gibt es Anwendungssituationen verschiedenster Natur.

In unserem Fallbeispiel aus Kapitel 3.1 besaß der Service Techniker neben einem stationären Rechner auch ein mobiles Gerät zur Verwaltung seiner Termine. Dieses Kapitel soll jetzt diese verschiedenen Plattformen für Kalendersysteme kurz vorstellen und jeweils ein bis zwei Fallstudien bringen, damit man sich die Kalendersysteme und deren Bedienung auf diesen Plattformen besser vorstellen kann.

In Kapitel 3 haben wir im Wesentlichen nur zwischen zwei verschiedenen Plattformen unterschieden: Zwischen stationären und mobilen Geräten. In beiden Kategorien gibt es jedoch noch weitere Unterkategorien von Geräten. Wir wollen in diesem Kapitel diese zwei Kategorien weiter nach unten aufbrechen.

Zunächst wollen wir uns die stationären Geräte anschauen. Es soll gezeigt werden, aus welchen Komponenten ein PC besteht. Danach sollen die Unterschiede zwischen einer Workstation und einem Desktop-PC herausgearbeitet werden. Die mobile Version dieser Geräte stellt der Laptop dar. Da auf beiden Systemen in der Regel die gleichen Betriebssysteme laufen und somit auch die gleiche Kalendersoftware eingesetzt werden kann, sollen diese beiden Geräte in einem Kapitel abgehandelt werden. Als konkrete Beispiele für Kalendersysteme auf diesen Plattformen sollen dann Microsoft Outlook und Mozilla Sunbird vorgestellt werden. Von diesen Anwendungen haben wir schon in Kapitel zwei Abbildungen gesehen.

Danach sollen die webbasierten Kalendersysteme beschrieben und dargestellt werden und gezeigt werden, was alles nötig ist, um auf diese zugreifen zu können. Auch hier sollen die Systeme anhand von zwei Fallstudien vertieft werden. Hier sollen die Webkalender von Freenet und Web.de vorgestellt werden. Auch diese Kalender kennen wir vom Aussehen her bereits aus Kapitel zwei.

In Kapitel 3.1 hatte unser Service Techniker einen PDA. Dieser Abschnitt soll erläutern was ein PDA ist und danach auch mittels einer Fallstudie ein Kalendersystem beschreiben, das auf diesen Systemen zu finden ist. An dieser Stelle wollen wir uns stellvertretend für alle PDAs den PalmOne Tungsten T5 anschauen.

Handys der heutigen Zeit werden immer moderner und bieten neben den Telefonfunktionalitäten längst weitere Features, wie Notizen oder Spiele. Daher ist es nicht verwunderlich, dass man mittlerweile auch Terminplaner auf diesen Systemen findet. Hier wollen wir uns das Handy Siemens U15 etwas näher anschauen, um etwas Einblick in diese Systeme zu bekommen.

Als letzte Plattformgruppe sollen die Smartphones vorgestellt werden. Vereinfacht gesagt ist dies der Versuch die Funktionalität von Handy und PDA in einem Gerät zu vereinen. Auch hier wollen wir uns eine Fallstudie anschauen. An dieser Stelle soll der Nokia Communicator 9500 beschrieben werden.

Abschließend soll eine kurze Diskussion noch die Vor- und Nachteile der verschiedenen Plattformen für die Kalendersysteme herausarbeiten.

## 4.1 Lokal auf dem Einzelplatzrechner oder Laptop



*Abb. 18 - Einzelplatzrechner DELL*

Sowohl in Unternehmen als auch im privaten Bereich ist der Personal Computer, kurz PC, nicht mehr aus dem Alltag wegzudenken. Wie der Name Einzelplatzrechner schon sagt, stehen diese Rechner meist einer Person zur Verfügung. Diese Rechner werden in Sachen Hardware und Software dann an die Bedürfnisse des Benutzers angepasst. Abb. 18 zeigt einen solchen Rechner. Aus Hardwaresicht bestehen diese Systeme aus den Komponenten zur: Eingabe von Informationen (Maus, Tastatur, Scanner, Webcam, ...), Speicherung von Informationen (Festplatte, CD/DVD-ROM, Diskette, ...), Internen Verarbeitung (Hauptplatine, CPU, Arbeitsspeicher, ...), Ausgabe von Informationen (Lautsprecher, Bildschirm, Drucker, ...), Kommunikation (Modem, Netzwerk, ...), und Schnittstellen (USB, FireWire, Serielle Schnittstelle, Parallele Schnittstelle, PCI, PCMCIA, ...) zum Anschließen von externen Komponenten. Aufgrund der vielen Schnittstellen, bieten diese Systeme oft optimale Erweiterungs- und Aufrüstmöglichkeiten. An dieser Stelle sollte der Begriff Einzelplatzrechner etwas näher beschrieben werden. Im privaten Bereich nennt man diese Systeme meist Heimrechner, in Unternehmen Arbeitsplatzrechner. Bei Arbeitsplatzrechnern unterscheidet man noch zwischen dem normalen Personal Computer und einer Workstation. Personal Computer sind meist zur Nutzung von Büroanwendungen vorgesehen. Darunter versteht man zum Beispiel Programme zur Textverarbeitung oder Tabellenkalkulation. Demnach wird auch eine nicht so hohe Rechenleistung benötigt. Workstations hingegen sind für komplexe Aufgaben, wie zum Beispiel im Bereich Grafik vorgesehen, die eine hohe Rechnerleistung erfordern. Die Grenzen zwischen PC und Workstation verwischen jedoch immer mehr, da selbst die privaten Heimrechner immer leistungsfähiger werden.



*Abb. 19 - Laptop von IBM*

Unter einem Laptop versteht man einen Einzelplatzrechner, der zum Transport und mobilen Einsatz optimiert wurde. Er besteht aus den gleichen Komponenten, und besitzt zusätzlich einen Akku, der den Betrieb über eine gewisse Zeitspanne auch ohne direkte Stromversorgung ermöglicht. Da Laptops möglichst leicht und klein sein sollen, wurden die Bauteile auf Größe hin optimiert. Abb. 19 zeigt einen Laptop. Auch hier unterscheidet man drei Gruppen: Laptop, Notebook und Subnotebooks. Notebooks haben in etwa die Größe eines Din-A4 Blattes, Subnotebooks sind noch kleiner und Laptops hingegen können auch größer als ein Din-A4 Blatt sein. Auch verwendet man für diese Geräte die Bezeichnung N-Spindel-Gerät. N gibt dabei die Anzahl der sich drehenden Laufwerke im Gerät an. Heute hat man meist ein DVD Laufwerk und eine Festplatte. Daher tragen die meisten Geräte die Bezeichnung Two-Spindel-Gerät.

Auf diesen Geräten kommt als Betriebssystem meistens Microsoft Windows zum Einsatz. Manchmal wird auch eine der vielen verschiedenen Distributionen von Linux verwendet. Auf Seite 11 zeigte die Abbildung den Terminplaner KOrganizer. Dieser ist ein Beispiel für einen Terminplaner unter Linux. Im Folgenden wollen wir uns jedoch auf Windows beschränken. Mit Hilfe von zwei Fallstudien sollen die Anwendungen Microsoft Outlook und Mozilla Sunbird vorgestellt werden, die wir schon in Form von ein paar Abbildungen im zweiten Kapitel gesehen haben. [WIKIAL] [WIKIPC] [WIKIWS] [WIKINB] [WIKILAP]

### 4.1.1 Fallstudie: Microsoft Outlook

Microsoft Outlook wird meist im Office Paket von Microsoft ausgeliefert, das noch die Tabellenkalkulation Excel, die Textverarbeitung Word und das Präsentationsprogramm Powerpoint beinhaltet. Das Office Paket gibt es in vielen verschiedenen Versionen. In den Premium Versionen ist auch noch die Datenbank Access enthalten. Outlook kann man wahlweise auch als Standalone Anwendung erwerben.

Outlook ist mehr als nur ein Terminverwaltungssystem. Seine eigentliche Stärke liegt in der E-Mail Funktionalität. Es ist über dieses Programm möglich, mehrere E-Mail Konten zu verwalten. Neben POP3-Mailservern werden auch IMAP-, HTML und Microsoft Exchange Server unterstützt. Des Weiteren beinhaltet Outlook auch eine Umfangreiche Datenbank zur Speicherung von Adressen und Kontakten. Diese Kontaktdaten lassen sich dann zur besseren Übersicht in Gruppen, wie privat und geschäftlich aufteilen. Man kann auch Notizen und Aufgaben ablegen. Outlook stellt noch eine Journalverwaltung zur Verfügung, die dabei hilft den Tagesablauf zu protokollieren.

Im Kalender lässt sich jede Art der bisher besprochen Form der Termine eintragen und anzeigen. Man kann diesen Terminen auch Personen zuordnen, die auf Wunsch sogar per Mail über den Termin informiert werden. Diese Personen kann man einfach im Adressbuch auswählen oder wahlweise neu eingeben. Das Adressbuch ist eine hilfreiche Komponente beim Schreiben von Mails, indem man dort den Empfänger auswählt.

Unten ist Microsoft Outlook 2003 zu sehen. In Kapitel zwei war die Vorgängerversion abgebildet. An der Basisfunktionalität hat sich wenig geändert. Im Wesentlichen wurde das Layout und die Bedienfreundlichkeit verbessert. [DOOOUT]

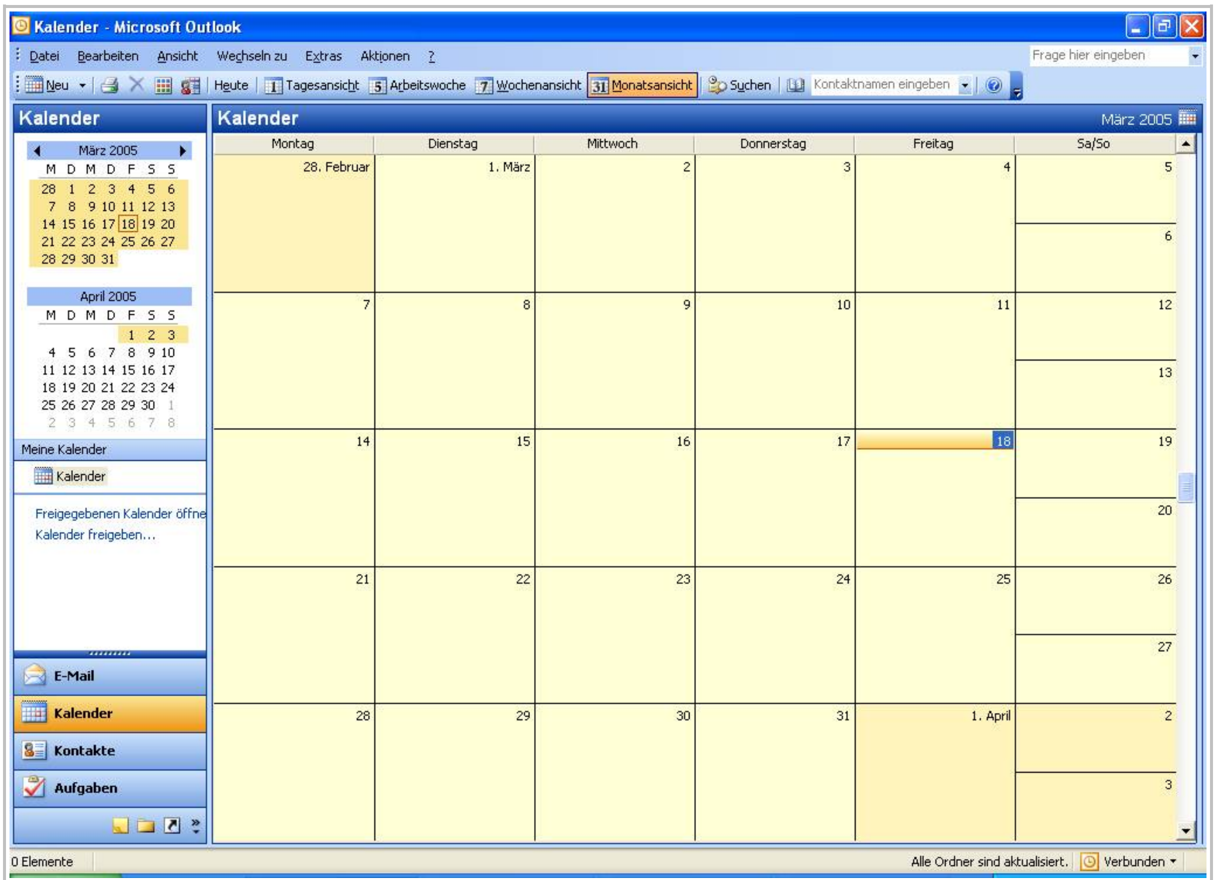



*Abb. 20 - Microsoft Outlook 2003 in der Monatsansicht*

### 4.1.2 Fallstudie: Mozilla Sunbird

Mozilla Sunbird basiert auf der Kalenderkomponente von Mozilla und wird von den Entwicklern komplett neu entworfen und programmiert. Sunbird gibt es aktuell in der Version 0.2. Die Versionsnummer zeigt, dass sich der Kalender noch in der Entwicklung befindet. Im Gegensatz zu Outlook, ist diese Anwendung frei verfügbar. Sie beinhaltet über die Kalenderfunktionalität hinaus jedoch keine weiteren Komponenten. Neben der Version für Windows gibt es den Terminplaner auch für Linux und Mac OS X.

Der Kalender soll möglichst klein gehalten werden und sich auf die wesentlichen Funktionen eines Kalenders beschränken. Viele andere Produkte der Mozilla-Reihe bieten eine Schnittstelle für Plugins. Jeder Anwender kann dann sein eigenes Plugin programmieren und dieses dann in die Anwendung integrieren. Plugins erweitern die Software um weitere Funktionalitäten. Es ist anzunehmen, dass auch Sunbird eine solche Schnittstelle erhalten wird. Somit kann jeder seinen Kalender erweitern und Anderen diese neuen Funktionen zur Verfügung stellen.

Im Moment lassen sich mit Sunbird Termine und Aufgaben verwalten. Sunbird bietet eine Tages-, Wochen-, Mehrwochen-, und Monatsansicht. Außerdem gibt es auch schon eine Erinnerungsfunktion für Termine.

Unten sehen wir einen Screenshot aus der Anwendung. Sie ist ähnlich wie Outlook aufgebaut: Oben ermöglicht eine Symbolleiste einen Schnellzugriff auf die wesentlichen Funktionen. Links findet man oben einen kleinen Kalender, der das Navigieren durch die Tage vereinfacht. Darunter hat man immer seine noch zu erledigenden Aufgaben im Blick. Rechts ist viel Raum für die Darstellung der Termine (Hier: Wochenansicht). [SUN02]

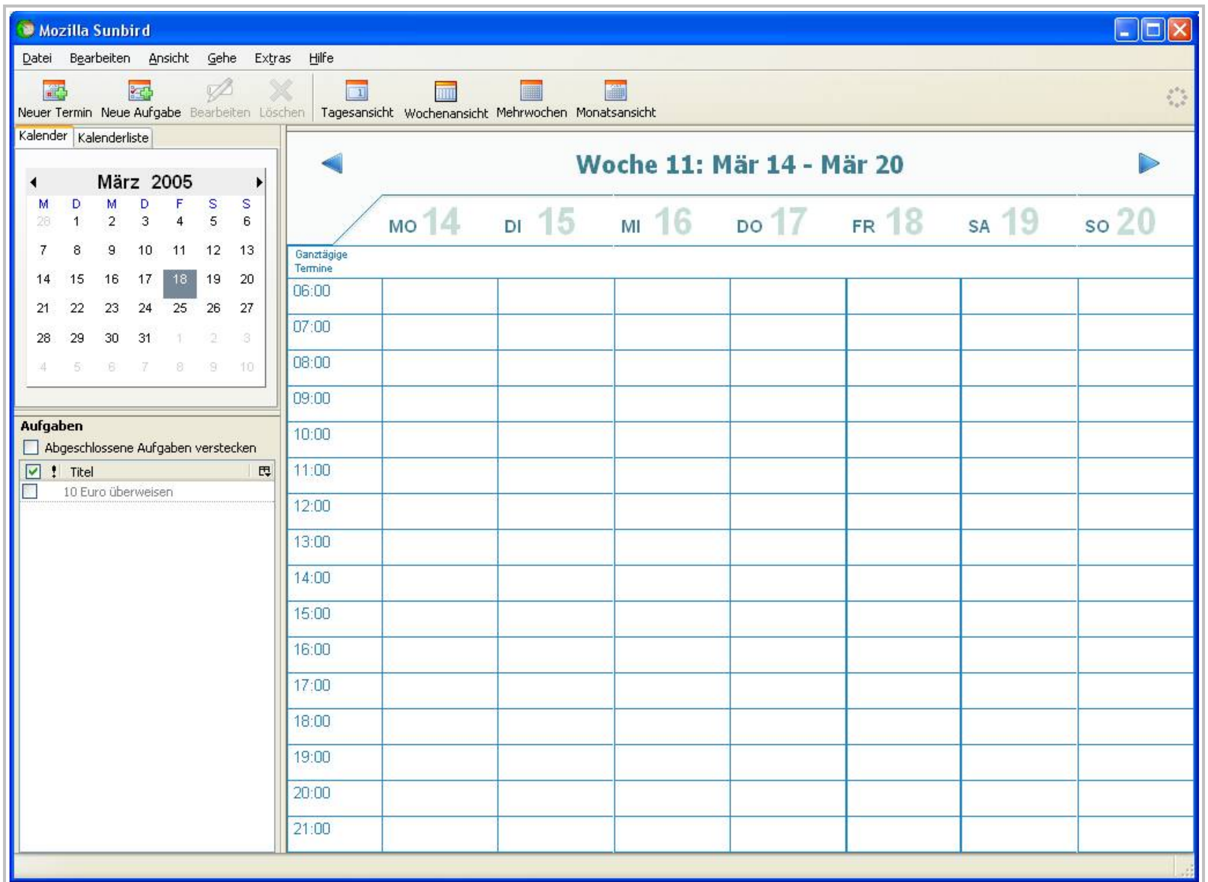



*Abb. 21 - Mozilla Sunbird in der Version 0.2 Hier: Wochenansicht*

## 4.2 Webbasierte Kalendersysteme

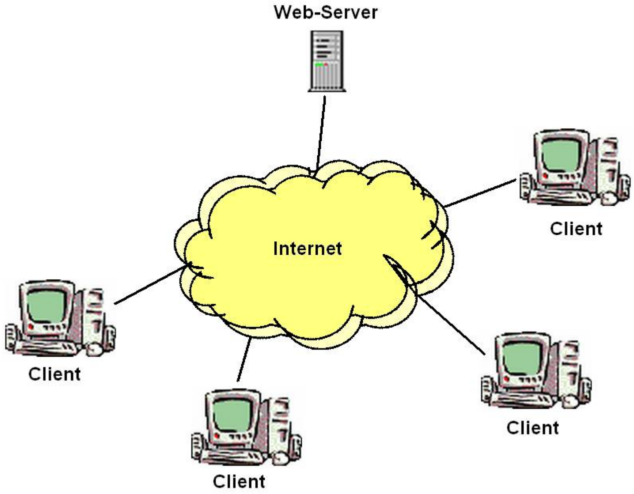


*Abb. 22 - Client-Server Prinzip*

Als nächste Plattform wollen wir uns das Internet anschauen. Das Internet funktioniert nach dem Server-Client Prinzip. Dies bedarf einer kurzen Erklärung. Unter einem Server versteht man einen leistungsfähigen Rechner, der Dienste für Andere bereitstellt. In einem Unternehmen mit einem großen Lager kann es zum Beispiel einen Server geben, der den aktuellen Lagerbestand der verschiedenen eingelagerten Teile speichert. Der zweite Agent im Server-Client Prinzip stellt der Client dar. Er nimmt Dienste dieser Server in Anspruch. In unserem Beispiel könnte unser Lagerverwaltungssystem Dienste wie Bestand überprüfen, Einlagern und Auslagern zur Verfügung stellen, die die Clientrechner des Unternehmens und somit deren Benutzer nutzen können. Server und Client sind in der Regel nicht ein und der selbe Rechner, sondern zwei Rechner. Auch bedient ein Server in der Regel mehr als nur einen Client. Diese Rechner kommunizieren dann entweder über ein firmeninternes Netzwerk, oder wie in Abbildung 22 gezeigt über das Internet. Zusammenfassend kann man also sagen, dass der Client die Benutzungsoberfläche oder die Benutzerschnittstelle der Anwendung anbietet. Die Funktionalität der Anwendung wird dann vom Server zur Verfügung gestellt.

Natürlich ist es dann nicht weit her geholt, dass man auf diese Weise auch einen Terminplanerdienst zur Verfügung stellen kann. Natürlich ist dann sowohl auf Server- als auch auf Clientseite spezielle Software nötig. Beim Beispiel des Terminplanerdienstes läuft serverseitig ein Webserver und clientseitig kommt ein Webbrowser zum Einsatz.

Ein Webserver ist ein Serverdienst, der Dateien bzw. Daten über das HTTP-Protokoll zur Verfügung stellt. Das Hypertext Transfer Protocol, kurz HTTP ist ein zustandsloses Datenaustausch-Protokoll zur Übertragung von Daten. Unter einem Webserver versteht man manchmal aber auch einen physisch vorhandenen Server, der einen HTTP-Dienst zur Verfügung stellt. Die Webserver liefern meistens HTML Seiten und deren Bilder und Stylesheets an die Clients zurück. Dank Skriptsprachen wie PHP, Perl oder ASP ist es auch möglich, statt den zuvor genannten statischen Daten auch dynamische Daten zurück zu geben. Hierdurch wird eine interaktive Benutzerführung ermöglicht. Alle Anfragen, die an den Server gestellt werden, werden in so genannten Logfiles mitgeschrieben, um z.B. Fehler oder einen Hackversuch zurückverfolgen zu können. Es gibt eine Reihe unterschiedlicher Webserver. Apache Webserver ist der Bekannteste.

Ein Browser, oder zu deutsch Stöberer, interagiert mit diesen Webservern. Er stellt die Anfragen des Benutzers an den Server und visualisiert dessen Antwortdaten auf dem Bildschirm des Benutzers. Mit dem Trend zur Multimedia sind heute Browser eine der zentralen Schnittstellen im Rechner. Man unterscheidet textbasierte- und grafische Browser. In der Zeit der Multimedia kommen fast nur noch grafische Browser zum Einsatz, die u.a. auch Bilder und Sounds wiedergeben können. Als Beispiele wären Microsoft Internetexplorer und Mozilla zu nennen. [WIKISC] [WIKISER] [WIKICL] [WIKIHTTP]

Nachfolgend werden Web.de und Freenet als Beispiele für diese Systeme vorgestellt.

### 4.2.1 Fallstudie: Kalender von Web.de

Web.de ist eine der bekanntesten Portalseiten im Internet. Sie bietet den Benutzern einen optimalen Start ins Internet. Viele aktuelle Informationen, wie zum Beispiel politische Ereignisse werden hier aufgelistet. Mit einem Klick kommt der Benutzer dann zu einem detaillierten Bericht. Web.de ist jedoch noch viel mehr. Es bietet einen sehr bekannten Freemaildienst an. Jeder kann sich eine kostenlose E-Mail Adresse, endend auf web.de, registrieren. Dieser Dienst wurde schon bei vielen Anbietervergleichen Testsieger. Web.de bietet seinen Kunden ein Webinterface zum Schreiben, Lesen und Verwalten der E-Mails. Derzeit hat jeder hierzu 12MB Speicherplatz zur Verfügung. Auch bekannt ist die Anwendung Smartsurfer von Web.de. Dies ist eine Anwendung, die speziell Modem und ISDN Nutzer als Zielgruppe hat. Vor jeder Einwahl zeigt diese Anwendung, welcher Provider gerade den günstigsten Tarif anbietet. Mit einem Klick kann man sich dann mit diesem Provider verbinden und los surfen. In der Liste befinden sich fast ausschließlich Provider, bei denen man sich nicht vorher extra anmelden muss. Das oben angesprochene Webinterface zur Verwaltung von E-Mails hat ebenfalls noch mehr Funktionen: Man kann unter Anderem auch SMS und MMS gegen Bezahlung verschicken und ein Adressbuch verwalten. Nicht verwunderlich ist es dann, dass Web.de auch einen webbasierten Terminplaner zur Verfügung stellt.

Ein Screenshot dieses Terminplaners ist in Abbildung 22 zu sehen, hier sehen wir gerade die Tagesansicht. Aber auch schon in den Abbildungen 10 und 12 haben wir diesen Terminplaner kennen gelernt. Anders als die bisher gezeigten Terminplaner beschränkt sich Web.de bei seinem Dienst nur auf die Basisfunktionalität eines Terminplaners. Auch kann man bei einem neuen Termin weniger einstellen, als zum Beispiel bei Outlook. Jedoch sind die Einstellmöglichkeiten vollkommen ausreichend.

Ansonsten ähnelt die Bedienung sehr den Terminplanern auf den Einzelplatzrechnern. Der Benutzer kann zwecks Datensicherung die Termindaten als CSV-Datei auf seinen Rechner exportieren. Dieses Format wurde bereits in Abschnitt 2.4 vorgestellt. Zu bemängeln ist hier, dass es keine Möglichkeit gibt die Daten in einem standardisierten Format, wie zum Beispiel iCalendar, zu speichern, um eine höhere Kompatibilität zu anderen Terminplanersystemen zu erreichen. Analog dazu funktioniert der Import von Daten aus Dateien.

In der kostenlosen Version kann man maximal 500 Termine speichern. Für jemanden, der viele Termine hat, ist das viel zu wenig. Um dieses Limit aufzuheben muss man auf die kostenpflichtige Version umsteigen, die noch weitere Features bietet, wie zum Beispiel Werbefreiheit und mehr Speicherplatz.

Aufgrund dieses Limits ist der Terminplaner der kostenlosen Version weniger interessant für den Benutzer. [DEWEB]

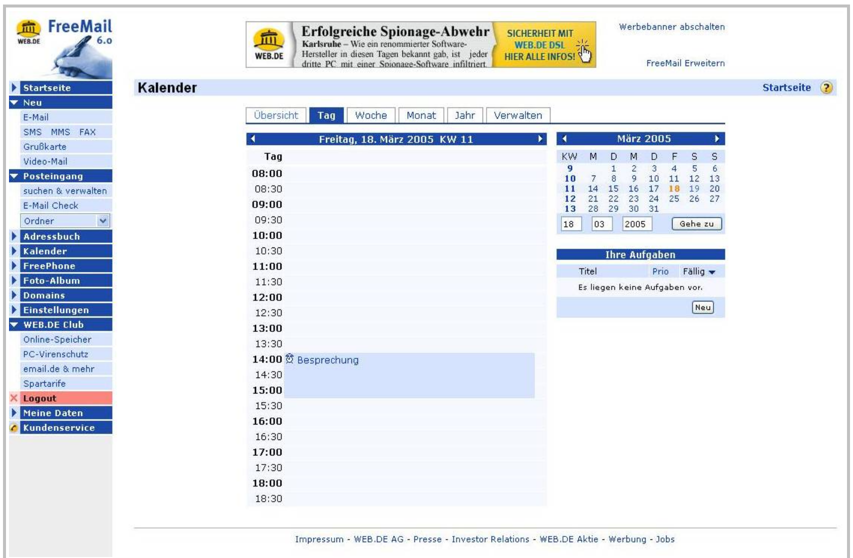


*Abb. 23 - Terminplaner von Web.de in der Tagesansicht*

### 4.2.2 Fallstudie: Kalender von Freenet

Freenet ist in erster Linie als Internet-Provider bekannt. Es bietet ein breites Sortiment an Tarifen für Modem, ISDN und DSL. Ähnlich wie Web.de bietet Freenet eine Portalseite an und seinen Kunden bietet es einen persönlichen Bereich an, in dem diverse Dienste wie SMS oder E-Mail zur Verfügung stehen. Um einen Account zu erhalten, muss man nicht einen Internetzugang über Freenet abonniert haben. Eine kostenlose Anmeldung genügt. Der Freemaildienst von Freenet ist bei weitem nicht so bekannt, wie der von Web.de. Auch Freenet bietet im Webinterface des E-Mail Dienstes einen Kalender zur Verwaltung der Termine an.

Der Terminplaner von Freenet ist wesentlich mächtiger, als der von Web.de. Unten sehen wir einen Ausschnitt des Terminplaners in der Tagesansicht. Freenet bietet wie alle anderen Terminplanersysteme alle die in Kapitel zwei aufgelisteten Basisfunktionalitäten. An dieser Stelle wollen wir auf die Sonderfunktionen eingehen. Wie wir im Screenshot unten sehen, sieht man auf jeder Seite die in das System eingetragenen Geburtstage. Außerdem kann man auch in eine Geburtstagsansicht wechseln, um eine Übersicht über alle Geburtstage zu bekommen. In Abbildung 15 wurde diese Ansicht bereits abgebildet.

In Kapitel 2.5 wurde auf Group Scheduling hingewiesen. Nicht jeder Planer unterstützt dieses Feature. Der Terminplaner von Freenet bietet dem Nutzer diese Funktionalität an.

Eine weitere Sonderfunktion ist ein umfangreiches Interface zum Druck der gespeicherten Termine. Hier kann der Benutzer unter Anderem einen Zeitraum festlegen, um die zu druckenden Termine auszuwählen.

Die Basisversion dieses Dienstes ist kostenlos. Hier sind im Gegensatz zu Web.de keine eng bemessenen Limits gesetzt. Einen enormen Nachteil gibt es jedoch: Man kann weder Termindaten importieren noch exportieren. Ob dies in der Premium Version möglich ist, ist in der Leistungsbeschreibung auf Freenet leider nicht vermerkt. [FREESCH]

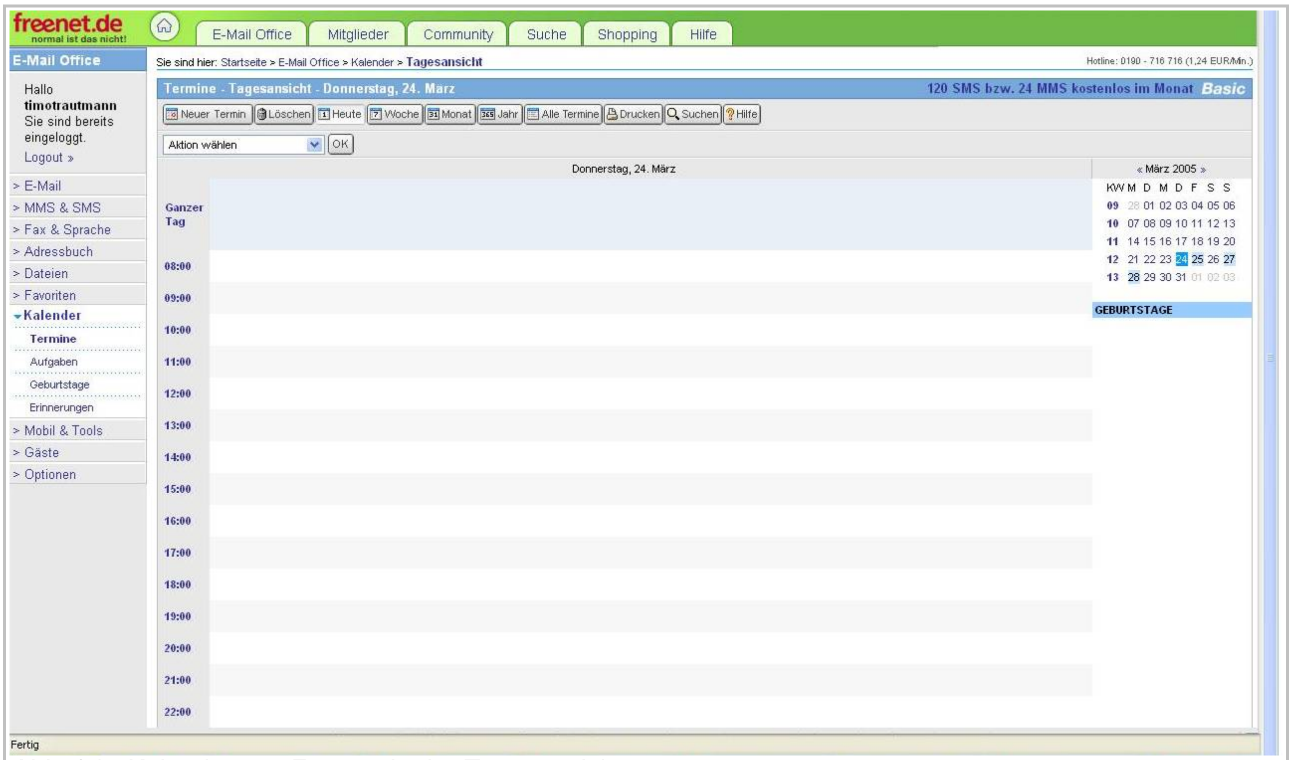



*Abb. 24 - Kalender von Freenet in der Tagesansicht*

## 4.3 Kalendersysteme auf dem PDA



*Abb. 25 - PalmOne*

Im zweiten Teil dieses Kapitels wollen wir nun unsere Aufmerksamkeit ganz auf Plattformen richten, die für Mobilität optimiert wurden. An dieser Stelle wollen wir mit dem PDA beginnen. Unter einem Personal Digital Assistant, kurz PDA, versteht man einen kleinen tragbaren Computer, der meist mit einem schnell startenden Betriebssystem ausgestattet ist. PDAs werden hauptsächlich für die persönliche Kalender-, Adress-, und Aufgabenverwaltung verwendet. Die neuen Geräte haben eine so hohe Leistung, dass sie auch als Navigationssysteme eingesetzt werden können.

PDAs werden standardmäßig mit sogenannter PIM-Software ausgeliefert (Personal Information Manager). Hierunter versteht man Adressbuch, Terminplaner, Kalender, Notizblock, Aufgabenplaner, E-Mail und Projektmanagement. Weitere Anwendungen, die man oft auf dem PDA findet, sind Textverarbeitung, Tabellenkalkulation, Rechner und Spiele. Dank externer Anschlüsse kann Software für den PDA aus dem Internet herunter geladen werden und danach zum Beispiel über USB auf das Gerät übertragen werden. Dank der hohen Leistung dieser kleinen Computer, verwendet man sie auch schon zur mobilen Datenerfassung. An manchen Stellen können sie dadurch den Standardcomputer ersetzen. Durch die Kombination von Bluetooth und WLAN mit einem Voice over IP Receiver (VoIP) wird es immer öfters möglich sein, das PDA als IP-Telefon zu verwenden. Da die Geräte über viele externe Anschlüsse verfügen, ist es kein Problem einen GPS-Empfänger anzuschließen. Hierdurch kann das PDA seine Position ermitteln und mit geeigneter Software als Routenplaner eingesetzt werden. Viele PDAs verfügen über einen Slot für Speicherkarten. So ist der Speicherplatz nahezu unbegrenzt. Man kann viel Software gleichzeitig nutzen, oder sogar Bücher auf dem PDA lesen.

An Anschlüssen findet man neben den kabelgebunden Anschlüssen USB und Seriell immer häufiger auch die kabellosen Anschlüsse Infrarot und Bluetooth. Diese ermöglichen eine Kommunikation der Geräte untereinander, sowie zwischen PC und PDA. Mittels geeigneter Synchonistationssoftware, ist es möglich, Daten zwischen PC und PDA abzugleichen. Dieser Vorgang wird meist als HotSync oder ActiveSync bezeichnet. Oft werden Daten wie E-Mails, Adressen und Termine abgeglichen. Dabei können Daten vom PDA auf den PC geladen werden und gleichzeitig Daten vom PC auf das PDA, so dass am Ende auf beiden Systemen die gleichen Daten vorhanden sind. Diesen Vorgang nennt man Synchronisation und vermeidet Datenredundanz. Es ist bei diesem Vorgang unter Anderem auch möglich, dass der PC eine Website aus dem Web herunter lädt und dann auf dem PDA speichert. So hat der Benutzer auf dem PDA immer die aktuelle Offline-Version der Seite zur Verfügung.

Es gibt für PDAs viele verschiedene Betriebssysteme, die meist herstellerspezifisch sind. Der Hersteller Newton liefert seine PDAs mit dem Betriebssystem Newton OS aus, das als erstes komplett in C++ geschriebene Betriebssystem bekannt ist, das speziell auf Speicher und Prozessorverbrauch optimiert wurde. Palm liefert seine Geräte mit dem Firmeneigenen Betriebssystem PalmOS aus. Auch Linux gibt es kostenlos für den PDA. Hier ist die Distribution OPIE wohl die Bekannteste. Einige PDAs tragen die spezielle Bezeichnung Pocket PC. Hierauf findet man das Betriebssystem Windows CE. In der aktuellen Version auch bekannt als Windows Mobile. [WIKIPDA]

Nachfolgend wollen wir uns den Terminplaner unter PalmOS anschauen.

### 4.3.1 Fallstudie: PalmOne Tungsten T5

Der Tungsten T5 hat einen Intel XScale Prozessor mit 416 MHZ und 256MB Arbeitsspeicher. Das Display verfügt über eine Auflösung von 320x480 Pixel mit einer Farbtiefe von 16 Bit. Ein weiteres Feature ist die Hintergrundbeleuchtung. Das Gewicht dieses Geräts beträgt gerade einmal 145g. In Abbildung 25 ist das Gerät zu sehen.

In Sachen Software wollen wir gleich auf den Terminplaner zu sprechen kommen. Rechts sehen wir die Tagesansicht. Oben kann man zwischen den Tagen wechseln. Um 13:00 findet eine Besprechung statt. Um 17:00 liegt ein Essen an. Die Bedienung erfolgt nicht wie bei dem PC über Maus und Tastatur, sondern über einen Stift. Wenn man damit auf einen Bereich des Bildschirms drückt, erkennen eingebaute Sensoren diese Stelle und rufen die gewünschte Aktion auf. Unten findet man eine Symbolleiste, die den Wechsel zu anderen Terminansichten ermöglicht. Über das Feld „Neu“ kann man einen neuen Termin anlegen. Wenn man einen Termin auswählt, kann man sich über „Details“ nähere Informationen besorgen. „Gehe zu“ bietet ein umfangreicheres Interface zum Tageswechsel.




*Abb. 26 - T5 Tagesansicht*

In der Wochenansicht, abgebildet in Abbildung 27, sieht man aufgrund des kleinen Displays keine Termindetails, wie Titel, Beginn oder Ende mehr. Der zeitliche Rahmen und eventuelle parallele Termine werden nur grob dargestellt. Beim T5 kann man jeder Kategorie eine Farbe zuordnen. Anhand der Farbe können wir in der Wochenansicht erkennen, welcher Kategorie der jeweilige Termin angehört. Rechts daneben sehen wir einen Teil der Oberfläche zum Anlegen eines neuen Termins. Aufgrund des kleinen Displays kann man die Wiederholung eines Termins auf einer extra Bildschirmseite einstellen. Termine verfügen auch hier über die in Kapitel 2.1 angesprochenen Attribute, wie die Abbildung zeigt.




*Abb. 27 - T5 Wochenansicht*




*Abb. 28 - T5 Neuer Termin*

In Abb. 29 sehen wir einen Synchronisationsdialog zum Abgleich der Daten zwischen PC und PDA. Wie man sieht, unterscheidet man drei Modi: Bei der Synchronisation werden die Daten, die auf einem Gerät geändert wurden, auf dem anderen Gerät aktualisiert. Bei „Desktop überschreibt Handheld“ werden alle aktualisierten Daten vom PC auf den PDA übertragen. Wenn es auch aktualisierte Daten auf dem PDA gibt, werden diese durch die Daten auf dem PC überschrieben. „Handheld überschreibt Desktop“ hat den gleichen Effekt in umgekehrter Art und Weise. Mit dem Menüpunkt „Keine Aktion“ kann man einen Datenabgleich ganz unterbinden. Diese Synchronistationssoftware läuft unter dem Betriebssystem Microsoft Windows und ist für den T5. [PALOST5]




*Abb. 29 - T5 Synchronisation PC-PDA*

## 4.4 Kalendersysteme auf dem Handy

Eine weitere mögliche Plattform für Kalendersysteme ist das Handy. Gerade hier hat sich in den letzten Jahren viel verändert. Längst nutzt man es nicht mehr nur zum Telefonieren. Fotografieren, spielen, und Musik hören machen diese Telefone zur Multimediastation. Daher gibt es heute auch Terminplanersoftware auf diesen Geräten.



*Abb. 30 - Entwicklung des Handys*

Ein Handy, oder zu deutsch Mobiltelefon, ist ein kleines tragbares Funk-Telefon. In Abbildung 30 sehen wir ganz links das Icom IC2E FM aus dem Jahre 1982 und daneben das Siemens S45 mit GSM aus dem Jahre 2004. Zum Größenvergleich sehen wir noch rechts eine Zigarettenschachtel. Wie zu sehen ist, sind die Handys viel kleiner und handlicher geworden. Dies wurde in erster Linie durch eine bessere Netzabdeckung ermöglicht. Durch die bessere Abdeckung wird eine geringere Sendeleistung benötigt und somit weniger Energie. Akku und Antenne konnten so verkleinert werden.

Ein Handy besteht, wie ein drahtgebundenes Telefon auch, aus Lautsprecher, Mikrofon, Bedienteil und Steuerung. Die Steuerung erfolgt meist durch einen Microkontroller. Des Weiteren haben alle neueren Telefone ein Display. In der neusten Generation gleichen diese Displays einem kleinen farbigen Computerbildschirm. Handys haben zusätzlich einen Funkteil und eine eigene Stromversorgung, meist in Form eines Akkus. Zum Betrieb eines Handys ist in der Regel eine SIM-Karte nötig. Diese dient zur Identifizierung des Teilnehmers innerhalb eines Mobilfunknetzes.

Auf manchen aktuellen Handys laufen Betriebssysteme, die auf Symbian OS aufbauen. Alle aktuellen Geräte verfügen über einen Interpreter für die Programmiersprache Java. Dies ermöglicht allen Programmierern neue Programme für diese Handys heraus zu bringen, die sich der Benutzer dann auf sein Handy laden kann. Dies können zum Beispiel Spiele sein. Solche Software gibt es sowohl kostenlos, als auch kostenpflichtig.

Heutige Handys funktionieren nach dem GSM-Standard. Sie benutzen Frequenzen um 900 MHz, bzw. 1800 MHz. Moderne Triband-Handys können auch auf 1900 MHz arbeiten. Dies verspricht eine bessere Verbindungsqualität. Diese Technologie kommen vor Allem in der USA zum Einsatz. Die nächste Generation von Handys wird Universal Mobile Telekommunications System, kurz UMTS, verwenden.

Wie in der Einleitung schon erwähnt, haben die neuen Handys noch weitere Funktionen. Kaum ein Handy wird mehr ohne integrierte Digitalkamera ausgeliefert. Jedoch sind die damit gemachten Bilder meist nur von mäßiger Qualität. Der Hobbyfotograf muss also weiter auf ein Stand Alone Gerät zurückgreifen. Diese Handykameras dienen meist für Momentaufnahmen. Gemachte Fotos lassen sich dann auf dem Handy mittels einer Multimedia Nachricht (MMS) an Freunde schicken. Sofern es das Handy unterstützt, können auch E-Mails versendet werden.

Ein paar wenige Geräte gestatten es sogar, Filme von maximal 30-60 Sekunden Länge aufzuzeichnen. Aber auch hier lässt die Qualität in der Regel sehr zu wünschen übrig. In manchen Geräten ist ein integrierter MP3-Player oder ein integriertes Radio zu finden.

Viele Handys verfügen über einen seriellen Anschluss, oder einen USB Anschluss. Moderne Handys haben eine drahtlose Verbindung, wie zum Beispiel Infrarot. Hierüber lassen sich Daten, wie z.B. Termindaten, mit einem PC synchronisieren. [WIKIHANDY]

### 4.4.1 Fallstudie: Siemens U15

An dieser Stelle wollen wir uns das Handy Siemens U15 ansehen. Das U15 ist eines der ersten Mobiltelefone, das Video-Telefonie und Video/Audio-Streaming unterstützt. Das Handy verfügt des Weiteren über die Multimediafunktionen Videorecorder, MP3-Player und Bild-/Video-Datentransfer. Das Siemens U15 ist mit 2 VGA Kameras mit 3-fach Zoom ausgestattet. Das Handy verfügt über Triband und kann schnell zwischen den Standards GSM und UMTS umschalten. Dieses Siemenshandy hat 64MB internen Speicher, der viel Platz für eigene Dateien und Fotos bietet. Zum Lieferumfang gehören sogar ein paar Spiele, wie die Handyversion des Strategiespiels Anno 1503. In Abbildung 31 ist dieses Handy abgebildet. Das Handy hat einen USB- und einen Infrarot-Anschluss. Somit gibt es also optimale Anschlussmöglichkeiten an andere Geräte, wie zum Beispiel einem PC. Auf diese Weise lassen sich Daten austauschen und synchronisieren. Nachfolgend wollen wir uns jetzt speziell den Terminplaner dieses Handys anschauen.



*Abb. 31 - Siemens U15*

Rechts sehen wir die Tagesansicht. Hier ist keine Zeitleiste zu sehen, wie wir sie sonst immer als typisches Merkmal der Tagesansicht kennen gelernt haben. Alle Termine sind in Form einer Liste aufgeführt, beginnend mit dem frühsten Termin des Tages. Termine ohne Zeitangabe werden ganz oben dargestellt. Der Wecker neben dem Beginn signalisiert einen eingestellten Alarm, also eine aktivierte Terminerinnerung. Wenn man einen Termin ausgewählt hat und auf „ZEIGEN“ klickt, erhält man weitere Details zu diesem Termin. Durch das mittlere Symbol unten lassen sich neue Termine eintragen, löschen und editieren. Außerdem kann man hierüber auch die Ansicht wechseln. Über „ZURÜCK“ kommt man zum vorherigen Bildschirm. In der Überschrift wird das Datum angezeigt, zu dem die angezeigten Termine gehören.




*Abb. 32 - U15 Tagesansicht*

In Abbildung 33 sehen wir jetzt die Monatsansicht auf dem U15. Das heutige Datum wird grau hervorgehoben. In unserem Beispiel ist das der 21. Das ausgewählte Datum wird in schwarz dargestellt. In unserem Beispiel ist das der 10. Mit der Auswahl „ZEIGEN“ wird in die Tagesansicht gewechselt und die Termine des Tages werden aufgelistet. Ein kleines schwarzes Dreieck symbolisiert, dass an diesem Tag ein oder mehrere Termine anliegen. Mit der „*“ Taste des Handys kommt man zum vorigen Monat und mit der „#“ Taste zum nächsten Monat. Dies kann man in der Titelzeile des Displays erkennen.




*Abb. 33 - U15 Monatsansicht*

Das Handy unterstützt alle gängigen Ansichten und Termintypen, die bereits in Kapitel zwei aufgezählt wurden.

Das Handy bietet eine Funktion „Sync“ über die Telefonbucheinträge oder Termindaten schnell synchronisiert werden können. Hierzu kann zum Beispiel die drahtlose Infrarot Verbindung genutzt werden. Wahlweise ist die Synchronisation aber auch über den integrierten USB-Anschluss möglich. [U15SIM]

## 4.5 Kalendersysteme auf Smartphones



*Abb. 34 - Nokia Communicator 9500*

Als fünfte und letzte Plattform in dieser Abschlussarbeit sollen die Smartphones vorgestellt werden. Smartphones sind der Versuch, die Funktionalität von Handy und PDA in nur einem Gerät zu vereinen. Smartphone heißt frei auf Deutsch übersetzt: „Schlaue Telefone". Mit Smartphones kann man sich also in Mobilfunknetze einloggen und telefonieren. Natürlich kann man mit Smartphones auch andere Mobilfunkdienste in Anspruch nehmen, die man auch auf dem Handy nutzen kann. An dieser Stelle wären als gängige Beispiele SMS und MMS zu nennen. Dies macht aber nur rund die Hälfte der Funktionalität eines Smartphones aus. Zur anderen Hälfte beinhalten diese Geräte die Funktionalitäten eines PDAs. Also Terminplanung, Notizen, Adressverwaltung, Textverarbeitung, Tabellenkalkulation und noch vieles mehr.

Hersteller für diese Geräte kommen aus beiden Branchen. Aus der Branche der PDAs ist PalmOne zu nennen, aus dem Bereich der Handys Nokia. Smartphones von Nokia gleichen vom Aussehen her eher einem Handy. Smartphones von PalmOne hingegen gleichen vom Aussehen her eher einem PDA. Das ist wenig verwunderlich, da der Ansatzpunkt dieser Geräte einmal vom PDA und einmal vom Handy ausging. Unterm Strich haben diese Geräte trotz des sehr unterschiedlichen Aussehens einen ähnlichen Funktionsumfang.

Auf Smartphones findet man also die PIM-Software, die wir auch auf den PDAs gesehen haben (Kalender, Adressbuch, Aufgabenliste). Aus dem Handybereich wurden Funktionalitäten, wie Kurzwahllisten und Telefonregister übernommen. Aber durch die Kombination beider Geräte ergeben sich auch neue Nutzungsmöglichkeiten. Mit einem integrierten Webbrowser und E-Mail Programm kann man jetzt dank des Netzanschlusses auch im Internet mobil surfen. Die Preise für solche Verbindungen liegen jedoch meistens jenseits von Gut und Böse. Aber für ein renommiertes Unternehmen kann dies durchaus eine Option sein.

Auf den Smartphones findet man meistens eins der drei folgenden Betriebssysteme vor: PalmOS, SymbianOS oder Windows CE. Im Bereich Windows findet man auch manchmal die speziellere Version Windows Powered Smartphone 2002, die kleine Version von Windows CE ohne Touchscreen.

In Abbildung 34 sehen wir als Beispiel für ein solches Gerät den Nokia Communicator 9500. Zusammengeklappt sieht er wie ein etwas größeres Handy aus. Wenn man ihn aufklappt sieht er wie ein kleines PDA aus. Bei der Synchronisation mit diesem Gerät kommt SyncML zum Einsatz. Dieser Standard wird im nächsten Kapitel vorgestellt.

In der folgenden Fallstudie wollen wir uns noch ein paar technische Daten dieses Gerätes, sowie den Terminplaner näher anschauen. [WIKISMART]

### 4.5.1 Fallstudie: Nokia Communicator 9500

Der Nokia Communicator 9500 hat Anschlüsse zu Highspeed-Datenverbindungen, wie Wireless-LAN, HSCSD und EGPRS. An Software beinhaltet er Büro-Anwendungen zum Bearbeiten von Dokumenten, Tabellen und Präsentationen, Internet Browser und E-Mail Client, Kontaktdatenbank, Kalender und Aufgabenliste mit PC-Synchronisation. Intern verfügt das Gerät über 80MB Speicher. Dank eines MMC-Speicherkartenslots kann der verfügbare Speicher aber schnell erweitert werden. Eine integrierte VGA-Kamera ermöglicht das Aufnehmen von Videos und das Anfertigen von Fotos. Auch dieses Smartphone unterstützt den Tribandbetrieb. Das Gerät beherrscht des Weiteren die Bluetooth Funktechnik. In Abbildung 34 ist dieses Gerät in zwei Ansichten dargestellt.




Abb. 35 - 9500 in der Wochenansicht

Auch auf diesem Gerät beinhaltet der Terminplaner die in Kapitel zwei angesprochene Basisfunktionalität. Rechts sehen wir die Wochenansicht. Die Wochentage sind in Form einer zweispaltigen Tabelle angeordnet. Wenn am Tag ein Termin statt findet, wird dessen Zeit und Titel angezeigt. Wenn es viele Termine an einem Tag gibt, können diese in dieser Ansicht nicht mehr alle auf einmal angezeigt werden. Bei ganztägigen Terminen, wie zum Beispiel einem Geburtstag, entfällt die Zeitangabe. Stattdessen wird ein kleines Symbol dargestellt.




Abb. 36 - 9500 Menü

Wie die Abbildung 36 zeigt verfügt das Kalendersystem, wie die Kalendersysteme auf dem PC auch, über ein umfangreiches Auswahlmenü. Hierüber lassen sich Termine anlegen, editieren und löschen. Man kann hierüber auch die Ansichtsart auf die Termine ändern.




Abb. 37 - Details zu einem Termin

In der nächsten Abbildung sehen wir die Detailansicht eines Termins. Aufgrund des kleinen Displays ist diese Ansicht mit Hilfe eines Reiters in mehrere Seiten untergliedert. So befinden sich Notizen, Erinnerung, Wiederholung und Status des Termins auf jeweils einer extra Seite.

In Abbildung 38 ist die Monatsansicht abgebildet. Auf dem Communicator ist diese etwas anders, als auf den anderen Systemen. Rechts findet man den Kalender des aktuellen Monats. Im Moment ist der 23. ausgewählt. Rechts daneben findet man eine Liste mit den Terminen, die an diesem Tag statt finden. In der unten rechts stehenden Abbildung sieht man die Tagesansicht. Da diese Ansicht von den zuvor gesehen Tagesansichten nur wenig abweicht, soll an dieser Stelle nicht näher darauf eingegangen werden. [S80AT]




Abb. 38 - 9500 Monatsansicht




Abb. 39 - 9500 Tagesansicht

## 4.6 Vor- und Nachteile der Kalendersystem-Plattformen

Nachdem wir nun einen recht ausführlichen Einblick in mögliche Systemplattformen für Kalendersysteme erlangt haben, sollen nun ein paar Vor- und Nachteile der unterschiedlichen Plattformen diskutiert werden. Um den Vergleich zu vereinfachen, möchten wir ähnliche Plattformen mit den gleichen Vor- und Nachteilen in Gruppen zusammenfassen: Die Funktionalität von Handy und PDA wurde in den Smartphones vereinigt. Alle drei Gerätetypen haben ähnliche Eigenschaften. Wir wollen sie in der Gruppe der mobilen Endgeräte zusammenfassen. Des Weiteren wollen wir die webbasierten Systeme, Einzelplatzrechner und Laptop jeweils als einzelne Gruppen behandeln. Obwohl Laptop und Einzelplatzrechner ähnliche Komponenten und Software haben, kann man sie nicht vergleichen, da Einzelplatzrechner für den stationären Einsatz und Laptops für den mobilen Einsatz entworfen wurden. An dieser Stelle wollen wir unsere Diskussion mit den Einzelplatzrechnern beginnen.

Diese Geräte verfügen meist über einen sehr großen Bildschirm. In der heutigen Zeit sind 19 Zoll Flachbildschirme oder 21 Zoll Röhrenmonitore keine Seltenheit mehr. Auf der großen Bildschirmfläche lassen sich demnach viele Informationen auf einmal darstellen. Dies bietet speziell bei Kalendersystemen einen guten Überblick. So ist es auch in gröberen Ansichten, wie der Wochenansicht, problemlos möglich, ein paar Details, wie zum Beispiel den Titel des Termins, darzustellen. Vorteilhaft sind auch die Geräte zur Eingabe von Informationen in den Einzelplatzrechner. Eine große Tastatur und eine Maus garantieren eine schnelle und angenehme Arbeit. Die stationären Rechner haben heute auch eine sehr hohe Rechenleistung. Dies spart auch Zeit, da Anfragen des Benutzers an das System schneller beantwortet werden können. Der Einzelplatzrechner verfügt über eine große Anzahl von Schnittstellen. So kann fast jedes beliebige Gerät daran angeschlossen werden. Wenn eine Schnittstelle fehlt, kann diese durch Einbau einer Erweiterungskarte schnell nachgerüstet werden. Somit bieten diese Geräte optimale Anschlussmöglichkeiten für andere Geräte, wie zum Beispiel die mobilen Endgeräte. Dies vereinfacht die kombinierte Nutzung mehrerer Plattformen. Jedoch haben diese Geräte auch einen enormen Nachteil: Wie der Name stationärer Einzelplatzrechner schon sagt, kann man diese Geräte nicht mal eben in die Hosentasche stecken und mitnehmen. Auch unvorteilhaft ist, dass diese Geräte eine gewisse Zeit zum booten brauchen, bis sie einsatzfähig sind. Abschließend kann man sagen, dass diese Systeme besonders für das ausführliche Planen der nächsten Woche oder der kommenden Wochen geeignet sind. Eben dann wenn man etwas Zeit hat und eine gute Übersicht benötigt.

Laptops ähneln vom Aufbau her den Einzelplatzrechnern, sind jedoch für den mobilen Einsatz gedacht. Im Wesentlichen beinhalten sie auch die oben beschriebenen Vor- und Nachteile. Der Bildschirm ist jedoch kleiner. Dennoch hat man trotz des etwas kleineren Bildschirms noch einen recht guten Überblick. Viele Laptops haben statt einer Maus eine so genanntes Touchpad. Dies erschwert die Bedienung der Geräte. Jedoch ist es problemlos möglich, eine Maus an einem USB Port des Geräts anzuschließen. Diese Geräte benötigen auch eine gewisse Zeit zum booten. Gerade wenn man viel unterwegs ist, muss man das Gerät öfters auf- und abbauen, was auch einige Zeit in Anspruch nimmt. Laptops verfügen zwar über eine eigene Stromversorgung durch einen Akku, jedoch kann man meist nur wenige Stunden ohne eigene Stromversorgung auskommen. Wenn der Akku ausfällt und keine Steckdose erreichbar ist, hat man keinen Zugriff mehr auf seine Daten. Die Rechnerleistung ist meist etwas geringer, als auf den Einzelplatzrechnern. Jedoch ist diese für Kalendersysteme vollkommen ausreichend. Kalendersysteme auf diesen Geräten sind bedingt für den mobilen Einsatz geeignet.

Als nächstes wollen wir uns die mobilen Endgeräte anschauen. Hierunter fallen Handy, PDA und Smartphones. Zuerst fällt das kleine Display und die Tastatur ins Auge. Auf Handys sind die Zahlen-Tasten noch mit je drei bis vier Buchstaben belegt. Wenn man gerade den vierten Buchstabe dieser Taste haben möchte, muss man diese vier mal drücken. Abhilfe schafft eine automatische Texterkennung. Wenn man „Hallo“ schreiben möchte, drückt man jede Taste nur einmal, auf der der gewünschte Buchstabe steht. Das System erkennt dann, welches Wort man vermutlich gemeint hat. Gerade bei exotischen Wörtern schlägt diese Erkennung fehl und man benötigt zur Korrektur jede Menge Zeit. Auch auf dem Palm ist die Eingabe von Daten mit Hilfe des Stiftes alles andere als komfortabel und schnell. Wir haben aber auch gesehen, dass Geräte, wie der Nokia Communicator, über eine vollständige Tastatur verfügen. Die Tasten sind jedoch sehr klein und gerade ungeübte Menschen haben am Anfang viele Schwierigkeiten damit. Das kleine Display dieser Gerätegruppe lässt es nur zu, eine sehr begrenzte Anzahl von Informationen gleichzeitig darzustellen. Dies ist besonders dann von Nachteil, wenn man sich Termine in einer gröberen Ansicht, wie der Wochenansicht, anschauen möchte. Hier ist es dann nicht mehr möglich einzelne Details zum Termin auf dem Bildschirm zu zeigen. Jedoch verfügen diese Geräte über schnell startende Betriebssysteme und sind sofort einsatzbereit. Aufgrund der Größe lassen sie sich auch einfach in die Hosentasche stecken. Mobile Endgeräte haben, wie Laptops auch, eine eigene Stromversorgung durch einen Akku. Dies birgt wiederum die Gefahr eines leeren Akkus. Solche Geräte werden meist unterwegs eingesetzt, wenn man einmal schnell seine nächsten Termine checken möchte, oder wenn man neue Termine eintragen und verwalten möchte. Mobile Endgeräte hat man praktisch in jeder Situation zur Hand.

Als letzte Gruppe schauen wir uns die webbasierten Kalendersysteme an. Diese kann man immer dann nutzen, wenn man über einen Internetzugang und einen aktuellen Browser verfügt. Das hat den Vorteil, dass sie nicht an ein Gerät oder an ein Betriebssystem gebunden sind. Sie sind also Plattformunabhängig. Da selbst neue mobile Endgeräte über Browser und Internetzugang verfügen, ist die Nutzung auch hier möglich. Je nachdem auf welchem Gerät man den Kalender verwendet, kommen die oben beschrieben Vor- und Nachteile dieser Gerätegruppe zum Tragen. Ein weiterer Nachteil ist, dass man zur Nutzung einen Internetzugang benötigt. Gerade bei einem mobilen Endgerät sind die Internetkosten sehr hoch. Wenn einmal kein Zugang vorhanden ist, hat man keinen Zugriff mehr auf seine Termindaten. Das Kalendersystem liegt auf einem Server im Internet. Wenn es sich um den eigenen Server handelt, entstehen auch hierfür weitere Kosten. Aber auch wenn man einen Premium Kalender-Dienst abonniert hat, muss man dafür zahlen. Dieser Server im Internet kann ausfallen, so dass man trotz verfügbarem Netzzugang keinen Zugriff mehr auf seinen Kalender hat. Diese Art der Kalendersysteme ist für Menschen gedacht, die immer über einen Internetzugang verfügen und des öfteren an Rechnern sitzen, die ihnen nicht gehören. Ein Einsatz dieser Systeme wäre auch denkbar, wenn auf den Geräten, an denen diese Personen arbeiten, verschiedene Systemumgebungen installiert sind.

Rückblickend kann man also sagen, dass jedes System seine Vor- und Nachteile hat. Diese Vor- und Nachteile lassen sich durch den kombinierten Einsatz dieser Plattformen aufheben. Normalerweise nutzt man eine Kombination aus stationärem Rechner und mobilem Endgerät zur Verwaltung seiner Termine. Jedoch möchte man dann keine Zeit beim Übertragen der Daten zwischen den Plattformen verlieren.

Welche Standards zur Übertragung von Informationen zwischen Kalendersystemen gebräuchlich sind, soll das Thema des folgenden Kapitels werden.

# 5 Synchronisation zwischen Kalendersystemen

Im vergangen Kapitel haben wir gesehen, dass Kalendersysteme oft in Kombination genutzt werden. In diesem Kapitel sollen ein paar Standards vorgestellt werden, die den Austausch von Informationen ermöglichen. In der Fachterminologie nennt man solch einen Vorgang Synchronisation. Was Synchronisation genau bedeutet, soll das folgende praktische Beispiel erklären. Hierzu kommen wir zu unserem Service Techniker aus Kapitel drei zurück:

Der Service Techniker verwaltete seine Termine, wie wir wissen, mit einem PDA und einem Einzelplatzrechner. Während eines Arbeitstags trägt er drei neue Termine auf seinem PDA ein. Bei zwei Terminen musste er den Zeitpunkt ändern und ein weiterer Termin wurde komplett gestrichen. Zuhause erhielt er per Group Scheduling zwei weitere Termine per E-Mail. Am Abend synchronisiert er seine Termindaten zwischen PC und PDA. Durch diesen Vorgang werden die drei neuen Termine auf dem PDA auf das Kalendersystem des PCs kopiert. Die Änderungen an den zwei Terminen auf dem PDA werden auch an den Terminen auf dem PC vorgenommen. Der Termin, der auf dem PDA gelöscht wurde, wird jetzt auch auf dem PC durch die Synchronisation entfernt. Auf dem PC befinden sich jetzt noch zwei neue Termine, die automatisch durch das Group Scheduling eingetragen wurden. Diese zwei Termine werden auf das PDA kopiert. Damit ist der Synchronisationsprozess abgeschlossen.

Zusammenfassend kann man also sagen, dass durch die Synchronisation alle Daten, die seit der letzten Änderung verändert wurden, auf dem anderen Gerät aktualisiert werden. Dabei werden die Änderungen auf beiden Geräten berücksichtigt. In Kapitel 4.3.1 haben wir auch gesehen, dass man manchmal die Möglichkeit hat den Datenabgleich nicht nur bidirektional (d.h. in beiden Richtungen) vorzunehmen. Man kann zum Beispiel nur die Daten vom PDA auf den PC übertragen.

In den nachfolgenden Abschnitten wollen wir diesen Synchronisationsprozess von der technischen Seite aus betrachten. In dieser Abschlussarbeit sollen drei verschiedene Standards besprochen werden: iCalendar, SyncML und CalDAV. iCalendar ist ein Standard zum Austausch von Kalenderinformationen und ist im RFC 2445 definiert. Termindaten werden oft in Dateien mit Hilfe des iCalendar-Formats gespeichert. Auch kommt dieser Standard beim Group Scheduling zum Einsatz, bei dem man Termindaten per E-Mail an andere Menschen übermittelt. SyncML ist ein plattformunabhängiger Standard zur Datensynchronisation. Beim Nokia Communicator 9500 war bei der Datensynchronisation von SyncML die Rede. CalDAV ist die Lösung, um über WebDAV auf Kalenderdaten zugreifen zu können. WebDAV definiert Standards, die auf dem HTML Protokoll aufbauen, um Dokumente und Dateien im Netzwerk verändern oder schreiben zu können. CalDAV ist somit eine Erweiterung zu WebDAV.

In den vergangen Kapiteln haben wir schon des öfteren von einem dieser Standards gehört. Daher ist es jetzt an der Zeit, sich diese Standards etwas genauer anzuschauen.

Zunächst soll zu jedem Synchronisationsstandard etwas allgemeines gesagt werden. Darunter fällt zum Beispiel, was der Standard überhaupt ist, die grobe Funktionsweise und Anwendungsgebiete dieses Standards. Danach wollen wir uns ein paar theoretische Grundlagen dazu vor Augen führen und abschließend, wo es sinnvoll ist, ein kleines praktisches Beispiel machen.

Beginnen wollen wir an dieser Stelle mit dem iCalendar Standard.

## 5.1 iCalendar

iCalendar soll das zentrale Thema des Kapitels Synchronisation werden. Das Format soll ausführlich vorgestellt werden und danach wollen wir zwei Termine in ein Kalendersystem eingeben und uns anschauen, wie diese dann im iCalendar Format aussehen. Um diese Beispiele besser verstehen zu können, sollen davor noch die wichtigsten syntaktischen Regeln des iCalendar-Formats in einem theoretischen Abschnitt erklärt werden. Natürlich ist der Rahmen der Bachelor-Arbeit zu gering, um auf jedes Detail einzugehen. Daher soll die Vorgehensweise auch hier allgemeiner Natur sein. Beginnen wollen wir an dieser Stelle mit einer allgemeinen Einführung in iCalendar.

iCalendar ist ein Standard zum Austausch von Kalenderinformationen und wird im RFC 2445 definiert. Dieser Standard ist auch noch unter dem kürzeren Namen iCal bekannt. iCalendar fand seinen Ursprung in der gleichnamigen Anwendung iCal, die auf dem Apple MacIntosh zu finden ist. In dieser Terminapplikation wurde mit iCalendar eine einfache Möglichkeit geschaffen, die Termindaten mit anderen zu teilen. Heute wird dieser Standard von einer großen Zahl von Terminplanersystemen unterstützt. Apples iCal, Lotus Notes, KOrganizer, Microsoft Outlook, Mozilla Sunbird und Novell Evolution sind nur eine kleine Auswahl der Kalendersysteme, die diesen Standard verstehen. Termindaten in diesem Format werden meistens durch E-Mails ausgetauscht und sind somit zentraler Bestandteil des Group Schedulings, das wir in einem der ersten Kapitel kennen gelernt haben.

iCalendar ist ein Dateiformat und kein Protokoll. Es bietet also die Möglichkeit, Termindaten strukturiert in Dateien abzulegen. Jedoch schreibt das Dateiformat nicht vor, was mit diesen Dateien zu tun ist. iCalendar ist also unabhängig von Transportprotokollen. Daher können diese Dateien zum Beispiel auf einem WebDAV Server verwendet werden, oder auch einfach auf einem normalen HTTP Webserver zum Download angeboten werden. Veranstaltungstermindaten werden zum Beispiel des öfteren in diesem Format zum Download angeboten, um diese dann einfach in das eigene Terminplanersystem übernehmen zu können. Zuvor haben wir gesehen, dass dieses Format auch beim Austausch von Termindaten per E-Mail zum Einsatz kommt. CalDAV ist jedoch noch ein Thema dieses Abschnitts. Hier soll dann etwas genauer dargestellt werden, wie das iCalendar Format auf WebDAV Servern zum Einsatz kommt.

Jedoch gibt es auch Transportprotokolle, die extra zum Transport des iCalendar Formats entworfen wurden: Das iCalendar Transport-Independent Interoperability Protokoll (iTIP) wäre an dieser Stelle zu nennen. Dieses Protokoll stellt eine Verbindung zwischen den Kalendernutzern und einem so genannten „Organizer“ her. Dieser „Organizer“ hat die Möglichkeit, in den Terminplanern der Nutzer Termindaten zu verwalten. Das Protokoll stellt ihm hierzu Methoden wie zum Beispiel publish, request, reply, add, cancel oder refresh zur Verfügung. Auch dieses Protokoll stellt ein weiteres Beispiel für das Group Scheduling dar. Da man viele Parallelen zu dem noch vorzustellenden SyncML Standard findet, soll auf dieses Protokoll nicht weiter eingegangen werden. Die genaue Definition dieses Protokolls findet man im RFC 2446. Ein weiteres Beispiel wäre das iCalendar Message-based Interoperability Protokoll, kurz IMIP. Dieses Protokoll legt fest, wie genau die iCalendar Daten über E-Mail ausgetauscht werden müssen. Auch an dieser Stelle wollen wir auf eine genaue Erläuterung verzichten. Weitere Informationen hierzu kann man im RFC 2447 finden.

iCalendar ist ein Produkt der Engineering Task Force Calendaring and Scheduling Working Group. Entwickelt wurde der Standard von Frank Dawson (Lotus) und Derik Stenerson (Microsoft). iCalendar basiert auf dem früheren vCalendar Standard, der vom

Internet Mail Consortium, kurz IMC, heraus gegeben wurde. Wenn von vCalendar die Rede ist meint man also einen älteren Standard, der durch iCalendar mit weiteren Funktionen erweitert wurde.

Abschließend wollen wir noch auf die verschiedenen Dateiendungen und deren Bedeutung eingehen: Alle iCalendar Dateien sind gemäß dem MIME Standard text/calendar kodiert. Multipurpose Internet Mail Extensions, kurz MIME, ist ein Kodierstandard, der den Aufbau und die Struktur von Internetnachrichten festlegt. Als Beispiel wäre der Aufbau von E-Mails zu nennen. Bei iCalendar unterscheidet man zwei verschiedene Dateiendungen: ics und ifb. Die Dateiendung ics kennzeichnet eine Datei, die Termindaten beinhaltet. Der Inhalt kann von einem bis zu praktisch beliebig vielen Termineinträgen reichen. Die Dateiendung ifb kennzeichnet Dateien in denen Informationen zur freien und zur belegten Zeit von Personen zu finden sind. Also wann eine Person Zeit hat und wann nicht. Im nun folgenden theoretischen Kapitel werden wir genauer darauf eingehen. [WIKIMIME] [WIKIICAL] [WIKIICALE]

### 5.1.1 Theoretische Grundlagen

Diesen Abschnitt wollen wir jetzt ganz dem Aufbau der iCalendar Dateien widmen. In jeder Zeile dieser Dateien steht ein Attribut-Wert Paar. Diese Attribute werden in verschiedenen Elementen gruppiert. Beispiele solcher Elemente wären: Termine, Aufgaben, Tagesprotokollierungen, freie und belegte Zeiten einer Person und Erinnerungen. Eine Verschachtelung dieser Elemente ist möglich. So kann zum Beispiel ein Terminelement ein Erinnerungselement beinhalten. Nach dem Motto Beispiele sagen mehr als tausend Worte wollen wir gleich zum ersten Beispiel übergehen und anhand dessen die Syntax von iCalendar kennen lernen. Alle nachfolgenden Beispiele sind dem RFC 2445 entnommen. Zuerst wollen wir uns ansehen, wie ein Termin im iCalendar Format gespeichert wird.

```
BEGIN:VCALENDAR
  VERSION:2.0
  PRODID:-//hacksw/handcal//NONSGML v1.0//EN
  BEGIN:VEVENT
    DTSTART:19970714T170000Z
    DTEND:19970715T035959Z
    SUMMARY:Bastille Day Party
  END:VEVENT
END:VCALENDAR
```

Jede Datei muss mit BEGIN:VCALENDAR beginnen und mit END:VCALENDAR abschließen. VCALENDAR ist das Herzstück der Datei, also das Top-Level Element. Der gesamte Inhalt, der sich zwischen diesen zwei Ausdrücken befindet, wird als "icalbody" bezeichnet. Alle anderen Elemente, wie in unserem Beispiel der Termin, sind Unterelemente von VCALENDAR. Jedes Element beginnt mit BEGIN und endet mit END. Dazwischen findet man Attribute und deren Werte. Diese Attribute dienen zum Beispiel dazu, die Eigenschaften eines Termins zu beschreiben.

Sogar das Top-Level Element hat Attribute: Das VERSION Attribut gibt an, welcher Versionsnummer von iCalendar das vorliegende Dokument zu grunde liegt. Das kann wichtig bei der Interpretation der Datei sein. Womöglich benutze man in einer älteren Syntax von iCalendar eine andere Notation. Durch das Attribut PRODID kann man erkennen, welche Terminapplikation die Datei erstellt hat.

In unserem Beispiel ist ein Termin gespeichert. Termine werden durch einen VEVENT Block gekennzeichnet. Der Termin beinhaltet eine Feier, die vom 14. Juli 1997 um 17:00 bis zum darauf folgenden Tag um 03:59 Uhr und 59 Sekunden statt findet. Der Beginn

wird durch das Attribut DTSTART gekennzeichnet und das Ende durch DTEND. Immer wenn von einer Zeitangabe die Rede ist, wird in iCalendar die gleiche Notation verwendet. Als Beispiel soll uns der Beginn des Termins ausreichen. 19970714T170000Z ergibt etwas anders formatiert 1997 07 14 T 17 00 00 Z. Jetzt kann man schon gut den oben genannten Beginn ablesen. Die Notation folgt also der Regel Jahr-Monat-Tag T Stunde-Minute-Sekunde Z. Wenn keine Zeitzone, zu der wir noch kommen werden, angegeben ist, wird als Zeitzone UTC angenommen. Die Koordinierte Weltzeit, engl. Coordinated Universal Time (UTC), ist die Zeit, von der die Zeiten in den verschiedenen Zeitzonen der Erde abgeleitet werden. Das SUMMARY Attribut beinhaltet den Titel des Termins. Ein mögliches DESCRIPTION Attribut würde die Beschreibung des Termins festlegen.

Als nächsten Schritt wollen wir uns anschauen, wie eine Aufgabe im iCalendar Format aussehen könnte.

```
BEGIN:VCALENDAR
  VERSION:2.0
  PRODID:-//ABC Corporation//NONSGML My Product//EN
  BEGIN:VTODO
    DTSTAMP:19980130T134500Z
    SEQUENCE:2
    UID:uid4@host1.com
    ORGANIZER:MAILTO:unclesam@us.gov
    ATTENDEE;PARTSTAT=ACCEPTED:MAILTO:jqpublic@host.com
    DUE:19980415T235959
    STATUS:NEEDS-ACTION
    SUMMARY:Submit Income Taxes
  END:VTODO
END:VCALENDAR
```

Aufgaben werden durch einen VTODO Block gekennzeichnet. Der Wert des Attributs DTSTAMP gibt das Erstellungsdatum der Aufgabe wieder. DUE gibt an, wann die Aufgabe erledigt sein muss und in STATUS wird vermerkt, wie der aktuelle Status der Aufgabe ist. Ob sie zum Beispiel noch erledigt werden muss. Die UID definiert einen persistenten, globalen und eindeutigen Identifier für das Element. ORGANIZER hält den Ersteller und Verantwortlichen des Ereignisses fest. Dies geschieht meist durch Angabe seiner E-Mail Adresse. ATTENDEE definiert alle Teilnehmer des Termins. Auch hier werden meistens deren E-Mail Adressen hinterlegt. Die SEQUENCE Nummer zählt die Anzahl der Änderungen an der Aufgabe seit deren Erstellung. SUMMARY ist uns schon aus dem vorigen Beispiel bekannt. Die Aufgabe fordert also dazu auf, eingehende Steuern bis zum Ende des 15.4.1998 einzureichen.

Da dies eine sehr wichtige Aufgabe ist, wollen wir uns daran erinnern lassen. Dies bringt uns zum nächsten Element des iCalendar Formats, die Erinnerung. Diese wird in einem VALARM Block festgelegt.

```
BEGIN:VALARM
  ACTION:AUDIO
  TRIGGER:19980403T120000
  ATTACH;FMTTYPE=audio/basic:http://host.com/pub/
                       audio-iles/ssbanner.aud
  REPEAT:4
  DURATION:PT1H
END:VALARM
```

Das ACTION Attribut gibt an, auf welche Art die Erinnerung erfolgen soll. In unserem Beispiel soll die Erinnerung auf akustische Art und Weise erfolgen. Das ATTACH Attribut gibt dann das Ziel der abzuspielenden Datei an. In Terminen können angehängte Dateien auch über das ATTACH Attribut gekennzeichnet werden. TRIGGER definiert, zu welchem Zeitpunkt der Alarm ausgelöst werden soll. REPEAT legt fest, wie oft die Erinnerung wiederholt werden soll und DURATION legt in diesem Zusammenhang fest, wie groß die Zeitspanne zwischen den Erinnerungen sein soll. In unserem Beispiel ist es eine Stunde.

```
BEGIN:VCALENDAR
  VERSION:2.0
  PRODID:-//ABC Corporation//NONSGML My Product//EN
  BEGIN:VTODO
    ...
    BEGIN:VALARM
      ...
    END:VALARM
  END:VTODO
END:VCALENDAR
```

Nun können wir die Erinnerung in unsere Aufgabe integrieren. Dies ist rechts zu sehen. Beim Termin und bei der Erinnerung handelt es sich um die gleichen Daten, wie zuvor gezeigt. Um das Beispiel möglichst übersichtlich zu halten, sind die Attribute des Termins und der Erinnerung mit ... abgekürzt. Dieses Beispiel soll die Möglichkeit des Verschachtelns von Elementen exemplarisch zeigen.

Wir haben gesehen, dass ein paar Terminapplikationen auch eine Möglichkeit bieten, den Tagesablauf zu protokollieren. Das haben wir zum Beispiel bei Microsoft Outlook gesehen. iCalendar bietet auch die Möglichkeit solche Daten in andere Systeme zu portieren. Viele Attribute sind gleich. Daher soll nur noch auf bisher unbekannte Attribute eingegangen werden. Solche Protokollierungen werden im VJOURNAL Element gespeichert.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VJOURNAL
  DTSTAMP:19970324T120000Z
  UID:uid5@host1.com
  ORGANIZER:MAILTO:jsmith@host.com
  STATUS:DRAFT
  CLASS:PUBLIC
  CATEGORY:Project Report, XYZ, Weekly Meeting
  DESCRIPTION:Project xyz Review Meeting Minutes\n
          Agenda\n1. Review of project version 1.0 requirements.\n2.
          Def. of project processes.\n3. Review of project schedule.\n
          Participants: John Smith, Jane Doe, Jim Dandy\n-It was
          decided that the requirements need to be signed off by
          product marketing.\n-Project processes were accepted.\n
          -Project schedule needs to account for scheduled holidays
          and employee vacation time. Check with HR for specific
          dates.\n-New schedule will be distributed by Friday.\n-
          Next weeks meeting is cancelled. No meeting until 3/23.
  END:VJOURNAL
END:VCALENDAR
```

Eigentlich sind uns nur zwei Attribute unbekannt. CLASS gibt an, welche Personen die Berechtigung zum Zugriff auf dieses Element haben. In unserem Beispiel sind keine Einschränkungen festgelegt. Das zweite unbekannte Attribut ist das CATEGORY Attribut. Schon bei der Einführung wurde erwähnt, dass Termine in Kategorien eingeteilt werden können. Bei VJOURNAL Elementen ist das nicht anders. Wir können diese auch in Kategorien einteilen. Welcher Kategorie das Element angehört wird durch das Attribut CATEGORY festgelegt.

```
BEGIN:VCALENDAR
  ...
  BEGIN:VFREEBUSY
    ORGANIZER:MAILTO:jsmith@host.com
    DTSTART:19980313T141711Z
    DTEND:19980410T141711Z
    FREEBUSY;FBTYPE=BUSY:19980415T133000Z/19980415T170000Z
    FREEBUSY:FBTYPE=FREE:19980314T233000Z/19980315T003000Z
    URL:http://www.host.com/calendar/busytime/jsmith.ifb
  END:VFREEBUSY
END:VCALENDAR
```

Im vorigen Abschnitt war die Rede davon in Dateien festzuhalten, wann eine Person Zeit hat und wann nicht. Dabei ist es nicht wichtig, was die Person tut, wenn sie keine Zeit hat. Hierfür ist das VFREEBUSY Element sehr hilfreich.

Über das FREEBUSY Attribut können Zeitintervalle definiert werden, in denen die Person Zeit hat oder nicht. Über das Attribut URL kann man eine Webadresse als Option für das Element angeben. In unserem Beispiel zeigt die Webadresse, von wo diese iCalendar Datei herunter geladen wurde. Der Ersteller muss nicht zwangsläufig auch die Person sein, deren belegte und freie Zeit im VFREEBUSY Element dargestellt wird. Weicht der Ersteller vom Betroffenen ab, kann der Betroffene über das ATTENDEE Attribut angegeben werden. Wenn diese Daten auf mehrere Personen gleichzeitig zutreffen, kann auch eine Liste von betroffenen Personen erstellt werden.

Abschließend müssen wir noch kurz das Problem der Zeitzonen besprechen. Wenn nicht anders angegeben, werden Beginn, Ende und sonstige Zeitangaben der Ereignisse einfach in der allgemeine UTC Zeitzone angenommen. Ein paar Anwendungen rechnen einfach die lokalen Zeiten in UTC um und speichern dann die Termine im iCalendar Format. Beim Import erfolgt dann wieder eine Umrechnung. Auf diese Weise erspart man sich die Angabe der verwendeten Zeitzone in der iCalendar Datei. Dennoch werden solche Elemente im iCalendar Format unter der Bezeichnung VTIMEZONE angeboten.

```
...
BEGIN:VTIMEZONE
   TZID:US-Eastern
   LAST-
MODIFIED:19870101T000000Z
   BEGIN:STANDARD
      DTSTART:19971026T020000
      RDATE:19971026T020000
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      TZNAME:EST
    END:STANDARD
    BEGIN:DAYLIGHT
      DTSTART:19971026T020000
      RDATE:19970406T020000
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      TZNAME:EDT
   END:DAYLIGHT
END:VTIMEZONE
...
```

Da wir uns nur auf das Wesentliche beschränken wollen und in der folgenden Fallstudie VTIMEZONE nicht verwendet wird, sollen hier nur ein paar allgemeine Sätze formuliert werden. Die Zeitzone spiegelt das Verhältnis zwischen Lokalzeit und UTC wieder. Man unterscheidet zwischen zwei Zeitbereichen: STANDARD und DAYLIGHT. Uns ist das besser als Sommer- und Winterzeit bekannt. Da bei beiden je unterschiedliche Verhältnisse zur UTC vorliegen, müssen daher beide definiert werden. Die Attribute der Elemente definieren die Relation zwischen Lokalzeit und UTC und vergeben den Elementen eindeutige Identifier.

Dies soll als keine Einführung in das Format iCalendar genügen. Abschließend sei nur gesagt, dass ein VEVENT natürlich auch Attribute, wie zum Beispiel URL, ORGANIZER, DESCRIPTION, oder DTSTAMP beinhalten kann. Analog ist es auch bei den anderen Elementen des iCalendar Standards. Wenn ein Attribut nicht angegeben ist, wird es als nicht vorhanden gewertet oder es wird ein Standardwert angenommen. Beispiel: Zeitpunkte werden bei fehlender Zeitzonendefiniton als UTC angenommen. Ein weiteres Beispiel wäre die Zuordnung eines Termins in eine Kategorie. Fehlt das Attribut CATEGORY, gehört der Termin auch keiner Kategorie an. Da uns der prinzipielle Aufbau einer iCalendar Datei geläufig ist, wollen wir in Mozilla Sunbird einen Termin eintragen und uns dann anschauen, wie dieser im iCalendar Format aussieht. Dies soll dann den Abschluss des recht umfangreichen Kapitels iCalendar dieser Abschlussarbeit darstellen. [WIKIICAL] [WIKIICALE] [WIKIUTC] [RFC2445]

### 5.1.2 Fallstudie: iCalendar - Format

In dieser Fallstudie wollen wir in Mozilla Sunbird einen Termin eintragen. Hierzu wollen wir auf unser Terminbeispiel aus dem zweiten Kapitel zurück kommen: Das Mittagessen in der Mensa. Wir planen, am Donnerstag, den 14.04.2005 von 12:30 bis 13:30 in der

Mensa zu essen. Als Ort geben wir Darmstadt ein. Da Darmstadt mehr als nur eine Mensa hat, werden wir in der Beschreibung des Termins konkreter: Wir wollen in der Mensa der Fachhochschule Darmstadt zu Mittag essen. Als URL vermerken wir die Homepage der FH. Des Weiteren wollen wir uns 15 Minuten vor Terminbeginn erinnern lassen und kennzeichnen den Termin als privat. Da unser Mittagessen garantiert statt findet, vermerken wir es als bestätigt. Außerdem ist der Termin sehr wichtig, daher weisen wir ihm die Kategorie Favoriten zu. Am nächsten Tag soll der gleiche Termin noch einmal zur gleichen Zeit statt finden. Wir erstellen also einen periodischen täglichen Termin. Da der Termin nur am darauf folgenden Tag statt finden soll, endet die Periode bereits am 15.04.2005. Den Termin haben wir am Vorabend des 14.04.2005 angelegt. Die nachfolgende Abbildung zeigt die Eingaben in die Benutzungsoberfläche.

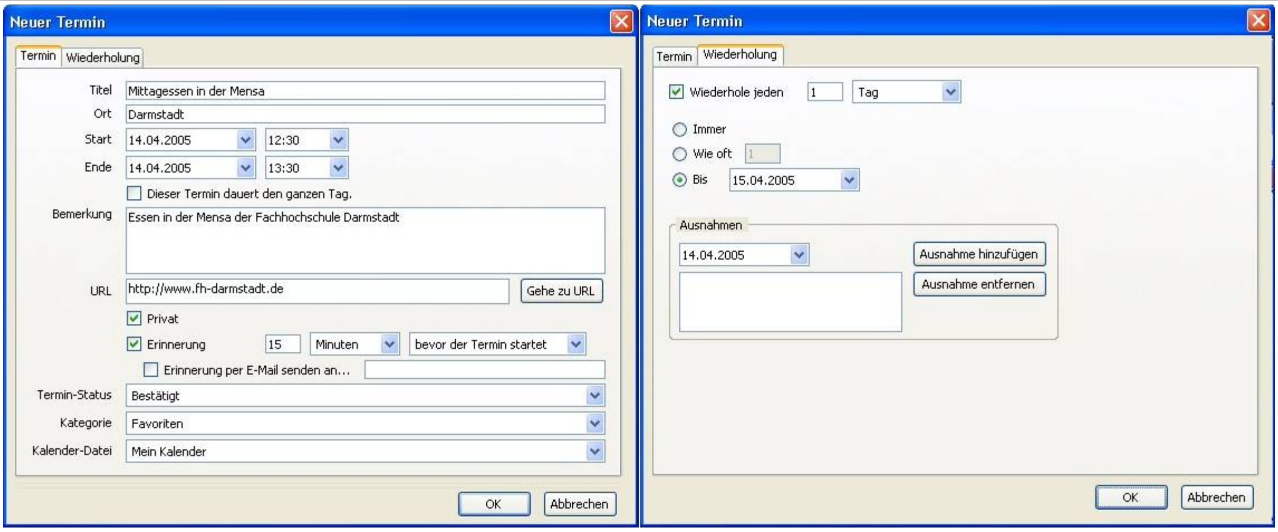



*Abb. 40 - Eingabe eines Termins in Mozilla Sunbird*

Im iCalendar Format identifiziert sich Sunbird als Dateiersteller über das PRODID Attribut. Das VEVENT Element kennzeichnet den Termin. Sunbird weist mit Hilfe des UID Attributs jedem Event einen eindeutigen Identifier zu. Titel und Beschreibung des Termins werden, wie wir bereits wissen, über die Attribute SUMMARY und DESCRIPTION gespeichert. Das LOCATION Attribut speichert den Ort des Termins. In unserem Beispiel ist das Darmstadt. Über CATEGORIES wird der Termin in die zuvor beschriebene Kategorie eingeteilt. Da wir den Termin als Privat eingestuft haben, wird dem CLASS Attribut der Wert PRIVATE zugewiesen. Der STATUS ist auf CONFIRMED, also bestätigt, gesetzt. Die Webadresse der Fachhochschule ist im Attribut URL gespeichert.

```
BEGIN:VCALENDAR
  VERSION:2.0
  PRODID:-//Mozilla.org/NONSGML Mozilla Calendar V1.0//EN
  BEGIN:VEVENT
    UID:1d0f55b0-ac4a-11d9-9411-c832996ed358
    SUMMARY:Mittagessen in der Mensa
    DESCRIPTION:Essen in der Mensa der Fachhochschule Darmstadt
    LOCATION:Darmstadt
    CATEGORIES:Favoriten
    URL:http://www.fh-darmstadt.de
    STATUS:CONFIRMED
    CLASS:PRIVATE
    RRULE:FREQ=DAILY;UNTIL=20050415;INTERVAL=1
    DTSTART:20050414T123000
    DTEND:20050414T133000
    DTSTAMP:20050413T182750Z
    BEGIN:VALARM
      TRIGGER;VALUE=DURATION:-PT15M
    END:VALARM
  END:VEVENT
END:VCALENDAR
```

Beginn, Ende und Erstellungsdatum wurden im theoretischen Kapitel bereits erschöpfend besprochen. Daher soll an dieser Stelle nicht weiter darauf eingegangen werden. Periodische Termine haben wir jedoch noch nie im iCalendar betrachtet. Hier erfolgt die Definition der Wiederholung über das PRULE Attribut. Die Wiederholung erfolgt DAILY, also täglich und im INTERVAL 1. Das heißt, dass der Termin jeden Tag statt findet und zum Beispiel nicht nur alle drei Tage. In diesem Fall wäre der Intervall drei. UNTIL kennzeichnet die Ende der Periode. Das Datumsformat von iCalendar ist uns schon geläufig. Die Periode endet also am 15.04.2005. Unser VEVENT Block beinhaltet ein verschachteltes VALARM Element. Das ist ja auch kein Wunder, da wir uns 15 Minuten vor Terminbeginn erinnern lassen wollten. TRIGGER gibt normalerweise den Zeitpunkt der Erinnerung an. Wenn der Zeitpunkt fehlt, wird der Beginn des Termins als Erinnerungszeitpunkt angenommen. DURATION hat den Wert -PT15M, also -15 Minuten. Somit ist es nachvollziehbar, dass die Erinnerung 15 Minuten vor Terminbeginn erfolgt.

## 5.2 SyncML

Nachdem wir ein Format zur Speicherung von Kalenderdaten kennen gelernt haben, wollen wir uns jetzt anschauen, wie die Daten zwischen verschiedenen Kalendersystemen, die womöglich noch auf verschiedenen Plattformen laufen, übertragen werden können. Ein möglicher Transferstandard ist SyncML. Zuerst soll das Konzept von SyncML heraus gearbeitet werden und danach wollen wir uns eine fiktive Situation vorstellen, in der wir SyncML als Lösung nutzen können. Bereits in Kapitel drei und vier haben wir gesehen, dass es immer mehr verschiedene Multimediageräte gibt und daher Datenaustausch immer wichtiger wird.

Die Synchronisation Markup Language, kurz SyncML, basiert auf XML und ist ein plattformunabhängiger Standard zur Synchronisation von Daten. Dabei beschränkt sich der Datenabgleich nicht nur auf Termindaten. Adressen, E-Mails und noch vieles mehr können ebenfalls über SyncML ausgetauscht werden. SyncML garantiert den Datenaustausch auch über Systemgrenzen und Plattformgrenzen hinweg. Daher ist SyncML ein offener Standard und jedem zugänglich, also nicht Herstellerspezifisch. Beim Datenaustausch hatte man das Problem, dass jeder Hersteller sein eigenes Verfahren hatte, Daten zu synchronisieren. Ein Beispiel wäre die Insellösung ActiveSync von Microsoft. SyncML soll dieses Problem lösen und den Datenaustausch auch zwischen Geräten verschiedener Hersteller sicher stellen. Der Standard ist des Weiteren nicht auf eine spezielle Netzwerkarchitektur festgelegt. Datenaustausch über das lokale Netzwerk oder das Internet ist genau so möglich, wie der Datenaustausch über das Mobilfunknetz, über WLAN oder einfach über ein USB Kabel. SyncML unterstützt dabei bereits etablierte Protokolle, wie zum Beispiel: HTTP, WAP, WSP, OBEX, Bluetooth und TCP/IP. Jedes Gerät, dass einen SyncML fähigen Client besitzt ist in der Lage, Daten mit einem SyncML Server abzugleichen. Typische Endgeräte, bei denen der Standard zum Einsatz kommt, wären unter anderem: PC, Mobiltelefon und PDA. In einer Fallstudie des vierten Kapitels haben wir beispielsweise gesehen, dass der Nokia Communicator 9500 diesen Standard unterstützt.

SyncML entstand durch eine Initiative der Unternehmen Nokia, Palm, IBM, Motorola, Ericsson, Symbian und Lotus. Diesen Zusammenschluss nennt man das SyncML Konsortium. Entwickelt wurde der Standard Anfang 2000. Space2Go war im Februar 2000 das erste deutsche Unternehmen, dass offiziell als Unterstützer des SyncML Konsortiums anerkannt wurde. Ein weiterer Hersteller für SyncML basierter Synchronisationstools wäre

das Schweizer Softwarehaus Synthesis. Space2Go bietet Tools an, die (neue und geänderte) Daten (z.B. Kalenderdaten) des Benutzers abholen und auf einem Server speichern. Auf einem anderen Endgerät kann er diese (neuen und geänderten) Daten mittels dieser Tools wieder herunter laden. Die Tools basieren komplett auf SyncML. Mittels dieser beiden Operationen, die auch automatisch ausgeführt werden können, wird die Synchronisation gewährleistet. Da die Daten nicht nur auf den Endgeräten, sondern auch auf einem Server abliegen, übernimmt SyncML auch die Rolle eines Backupservers. Wenn ein SyncML fähiges PDA defekt ist, sind die Daten schnell vom Server auf ein neues SyncML fähiges Gerät übertragen. Der Datenverlust beschränkt sich dann nur auf die Zeitspanne seit dem letzten Datenabgleich mit dem Server. Heute verwenden über 250 Unternehmen diesen Standard.

Für viele Menschen ist die Möglichkeit Daten und Einstellungen auf ein neues Gerät zu übertragen ein wichtiges Kriterium für den Kauf einer neuen Technologie. Das Übertragen der Daten vom alten Gerät auf das neue Gerät soll möglichst schnell, einfach und vollständig sein. SyncML ist, wie wir gesehen haben, eine Lösung für dieses Problem.

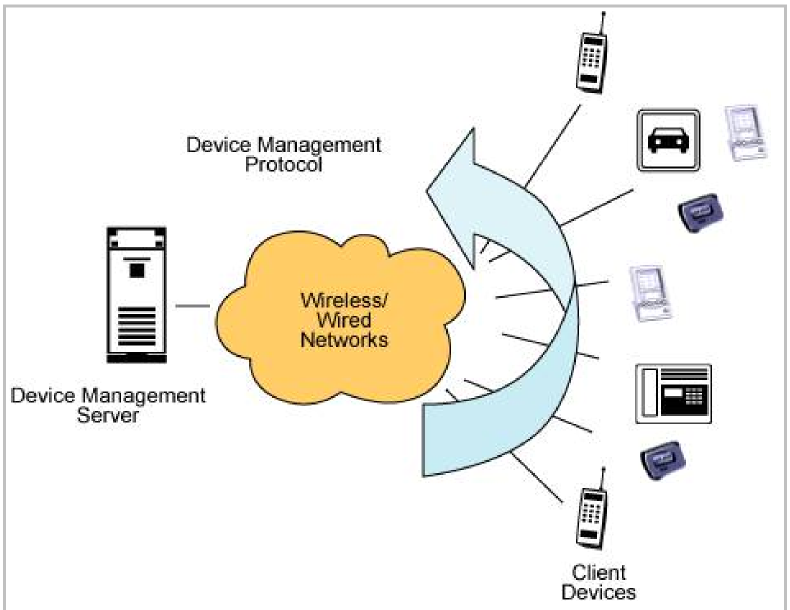



*Abb. 41 - Funktionsweise von SyncML*

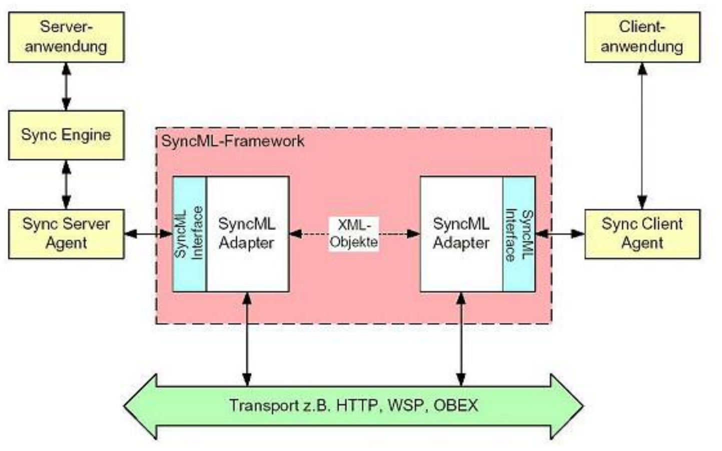



*Abb. 42 - Funktionsweise von SyncML - Komponentenansicht*

Wie man aus dem vorigen Text schon erahnen konnte, arbeitet das SyncML Protokoll nach dem Server-Client Prinzip. Die Clients sind die Endgeräte, deren Information immer Konsistent sein soll. Der Server stellt diese Konsistenz sicher. Dabei ist es egal, um welche Endgeräte oder Hersteller es sich handelt. Somit ist dieses System zentralistisch: Alle Daten, die synchronisiert werden sollen, werden auf dem Server gespeichert. Wenn auf einem Gerät eine Dateioperation statt findet, wird diese Änderung auf den Server übertragen. Dieser gibt dann die geänderten Daten an alle anderen Geräte weiter, die ebenfalls diese Datensätze beinhalten. Ein solcher Abgleich ist theoretisch in Echtzeit möglich. Hierzu werden wir aber im nächsten Abschnitt zu sprechen kommen. Die rechts oben stehende Abbildung erläutert dieses Prinzip. Die rechts stehende Abbildung erläutert SyncML aus technischer Sicht. Also welche Komponenten dabei zum Einsatz kommen. Das SyncML Interface überträgt die Daten zwischen Server und Client mittels herkömmlicher Protokolle, die zuvor genannt wurden und leitet sie zu den SyncML

Schnittstellen von Server und Client weiter. Übertragen werden die Daten mit Hilfe von XML. Der SyncAgent auf Clientseite ist zuständig für den Start einer Synchronisation. Des Weiteren verwaltet er die Übertragungsvorgänge auf der Clientseite. Auf Serverseite nimmt der Agent Synchronisationsanforderungen entgegen. Auf Serverseite gibt es zusätzlich noch eine Engine, die die Anforderung analysiert. Hierbei stellt sie fest, welche Daten zu verändern sind: Neue Datensätze, die der Client dem Server übermittelt hat, werden in die interne Datenbank übernommen. Datensätze in der Datenbank, die neuer sind als die des Clients, werden an ihn zurück gesendet. Dies ist jedoch nicht immer der Fall. Hierzu kommen wir jedoch noch später.

Auf diese Weise lässt sich im Prinzip eine beliebig große Datenmenge verwalten. Das Limit ist nur auf die vorhanden Ressourcen der beteiligten Geräte beschränkt. SyncML berücksichtigt dabei auch die begrenzten Ressourcen von mobilen Geräten. Beispiel: Die Synchronisation zwischen zwei Rechnern und einem PDA mittels SyncML wäre gegeben. Zwischen den beiden Rechnern wird eine große Datenmenge konsistent gehalten. Das PDA besitzt jedoch nicht so viel Speicherplatz. Somit ist es möglich die Synchronisation mit dem PDA auf das Adressbuch und die Termine zu beschränken. Andere größere Dateien, die unter den PCs ausgetauscht werden, werden nicht auf das PDA übertragen. SyncML besitzt beispielsweise Templates zum Austausch von Kalenderinformationen und Adressbuchinformationen. Jedoch kann man SyncML problemlos durch weitere Templates erweitern und somit praktisch jeden Datentyp synchronisieren. Auch ist positiv anzumerken, dass SyncML nicht ein komplett neues Protokoll ist, sondern auf bestehenden Internet und Webtechniken aufbaut.

Abschließend wollen wir noch auf die verschiedenen Synchronisationsverfahren eingehen. *Two way sync*: Hierbei tauschen Server und Client die veränderten Daten aus. Dies ist der gängigste Synchronisationstyp. Bei einem *Slow Sync* sendet der Client alle Datensätze an den Server. Dieser aktualisiert die Datensätze und sendet sie komplett wieder zurück. Bei einem *one way sync from client only* sendet der Client nur die aktualisierten Datensätze an den Server, ohne dass dieser neue Datensätze an den Client zurück schickt. Bei einem *refresh sync from client only* sendet der Client alle Datensätze an den Server. Dieser aktualisiert daraufhin seine Datenbank. Der Server sendet nichts zurück. O*ne way sync from server only* und *refresh sync from server only* sind die analogen Gegenstücke. Entweder schickt der Server nur aktualisierte Daten oder alle Daten an den Client. Der Client sendet in beiden Fällen nichts zurück. Mittels eines *Server Alerted Sync* kann der Server dem Client mitteilen, dass ein Synchronisationswunsch vorliegt. Wenn ein anderes Gerät aktuelle Daten auf den Server überträgt, kann der Server so den anderen Geräten mitteilen, dass aktualisierte Datensätze vorhanden sind. [WIKISYNC] [SP2G] [TECSYNC]

### 5.2.1 Fiktives Einsatzbeispiel für SyncML

In diesem Abschnitt wollen wir nochmals auf unseren Service Techniker zurück kommen. Stellen wir uns vor, dass nicht nur er selbst, sondern auch sein Unternehmen Termine festlegt. Des öfteren kommt es vor, dass er unterwegs ist und dort neue Termine in sein PDA einträgt. Zwischenzeitlich legt die Sekretärin des Unternehmens neue Termine für ihn fest. Wenn er abends zum Unternehmen zurück kehrt, stellt er fest, dass die Termine, welche die Sekretärin vereinbart hat sich mit den Terminen, die er über den Tag hin in sein PDA eingetragen hat, überschneiden. Es kommt zu zeitintensiven Diskussionen und manchmal sogar zu Streitigkeiten. Hier wäre eine SyncML Server-Client Lösung sehr hilfreich. Jedoch ist sie nur sinnvoll, wenn der Synchronisationsprozess in Echtzeit erfolgt.

Also sobald entweder die Sekretärin einen Termin anlegt oder er in seinem PDA etwas festhält, sollte eine komplette Synchronisation erfolgen. Dabei reicht es nicht aus, dass nur die aktualisierten Daten an den Server gesendet werden. Der Server muss die aktualisierten Daten auch sofort an alle beteiligten Endgeräte weiterleiten. So wüsste die Sekretärin sofort Bescheid, wenn er einen Termin in sein PDA eingibt. Es käme zu keinen Fehlplanungen mehr. Auch wenn der Service Techniker unterwegs ist, sind Unternehmen und er immer auf dem aktuellsten Stand der Dinge. Dies setzt natürlich eine ständige Verbindung zum SyncML-Server voraus. Wenn der Service Techniker unterwegs ist, könnte die Verbindung zum Beispiel über ein Mobilfunknetz bestehen. Je nachdem wie viele Synchronisationsprozesse täglich nötig wären, könnte sich dieser Komfort in deutlich höheren Kosten niederschlagen. Des Weiteren muss man beim Umstieg auf eine solche Technologie zusätzlich noch einen Server stellen. Egal ob gemietet oder gekauft, auch der Server verursacht weitere Kosten. Dennoch ist es für besonders mobile und große Unternehmen eine Möglichkeit, zu der man sich Gedanken machen sollte. Aber auch der Heimanwender kommt mit SyncML beim Austausch seiner Daten in Kontakt. Jedoch tritt dabei der Aspekt der Echtzeit in den Hintergrund.

## 5.3 CalDAV

Als dritte und letzte Technologie wollen wir uns CalDAV ansehen. CalDAV ist auch ein Verfahren zum Austausch von Kalenderinformationen. Dieser Standard ist eine Erweiterung von WebDAV, die es ermöglicht, Kalenderinformationen über einen WebDAV Server auszutauschen. Um CalDAV besser verstehen zu können, sollten wir an dieser Stelle klären, was man unter WebDAV versteht.

### 5.3.1 Einführung in WebDAV

WebDAV ist eine Arbeitsgruppe der Internet Engineering Task Force, kurz IETF. Diese Arbeitsgruppe wurde 1996 ins Leben gerufen. Ziel dieser Gruppe ist es, einen neuen Standard zu schaffen, der auf dem HTTP Protokoll basiert und es ermöglicht, Dateien im Netzwerk zu verändern und zu schreiben. HTTP wurde ursprünglich nur als Leseprotokoll konzipiert. WebDAV soll das Protokoll um eine Schreibenfunktionalität erweitern. Historisch gesehen war WebDAV mit Sicherheit das erste Protokoll, dass XML unterstützte.

WebDAV beinhaltet neue Methoden und Header für das HTTP Protokoll. Darunter sind neue Methoden, wie zum Beispiel: PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK. PROPFIND dient zum Auslesen von Verzeichnissen eines entfernten Servers. Des Weiteren kann man mit dieser Methode weitere Informationen über Dateien in Erfahrung bringen. Über PROPPATCH kann man diese Dateiinformationen verändern. MKCOL ermöglicht die Verzeichniserstellung. Über COPY und MOVE kann man Dateien kopieren oder verschieben. Über LOCK und UNLOCK kann man das entsprechende Ziel sperren oder wieder freigeben. Über eine URI wird dabei das jeweilige Ziel, also die Datei, auf die die jeweilige Operation angewendet werden soll, adressiert. Eine URI ist ein eindeutiger Identifier für eine Ressource (z.B. eine Datei).

Die Vorteile von WebDAV liegen auf der Hand. Jeder weiß bestimmt, wie mühselig es ist, Dateien über ein HTML Formular hochzuladen. Jede Datei muss einzeln hoch geladen werden. Mit WebDAV wird es einfacher: Sogar ganze Verzeichnisse können mit nur

einem Klick vom lokalen Rechner auf den Server übertragen werden. WebDAV ist ein offenes und recht verbreitetes Protokoll.

Heute unterstützt eine große Zahl von Applikationen diesen Standard. Als Beispiele wären Apache, iCal, Microsoft IIS und Kiwi zu nennen. Selbst für WebDAV gibt es heute eine große Zahl von Erweiterungen. Eine davon ist CalDAV, die wir uns um folgenden Abschnitt betrachten wollen. [WIKIDAV]

### 5.3.2 Einführung in CalDAV

Spezifikationen zum CalDAV Standard wurden im Jahre 2003 erstmalig durch Lisa Dusseault der IETF vorgelegt. Schnell stieß dieser Vorschlag auf breite Zustimmung bei den Kalendersystemherstellern. Im Januar 2005 begann man mit den ersten Tests dieses Standards. CalDAV wurde nicht von der IETF entwickelt. Jedoch ist es geplant, CalDAV als IETF Standard zu publizieren. CalDAV soll unter allen Plattformen und Betriebssystemen zur Verfügung stehen, auf denen man das Bedürfnis hat, Termininformationen auszutauschen. Auch CalDAV soll frei verfügbar sein, um den herstellerspezifischen Insellösungen entgegenzuwirken.

Das Konzept von CalDAV besteht darin, Termindaten als HTTP basierende Ressourcen zur Verfügung zu stellen. Dabei sollen die Daten auf dem Server im iCalendar Format gespeichert werden. Somit ist gewährleistet, dass jeder Webbrowser die Termindaten in einem standardisierten Format herunter laden kann. CalDAV soll durch dieses Konzept eine Synchronisationsmöglichkeit für Kalenderdaten bereitstellen. WebDAV und die Fähigkeit iCalendar Dateien zu verarbeiten, wären als Voraussetzung für die Nutzung von CalDAV auf einem Server zu nennen. Neben personalisierten Kalendern lassen sich auch öffentliche Kalender anlegen, die jedem zur Verfügung stehen. Dies ist zum Beispiel bei Vereinsterminen sinnvoll. [WIKICAL]

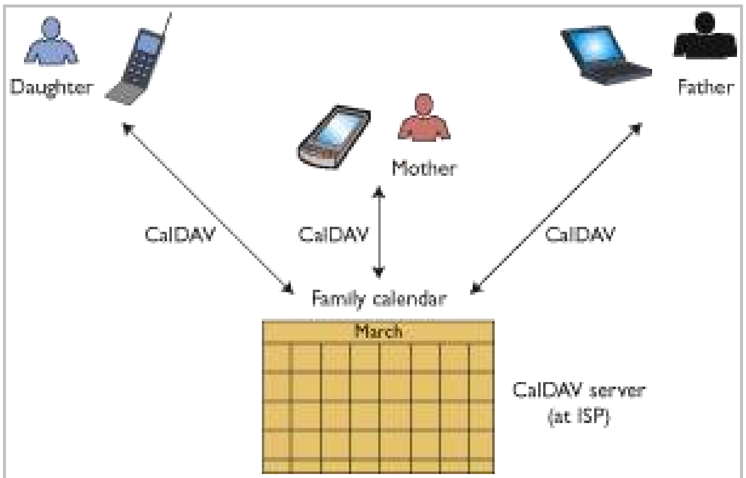


*Abb. 43 - Konzept von CalDAV*

Die Funktionsweise von CalDAV folgt ebenfalls dem Server-Client Prinzip. Da wir dieses Konzept im vorigen Kapitel schon ausführlich kennen gelernt haben, wollen wir dieses Konzept hier schnell abhandeln. Schauen wir uns als Beispiel die rechts stehende Abbildung an. Eine Familie möchte zentral ihre Termindaten verwalten. Der Vater nutzt einen Laptop, die Tochter ein Handy und die Mutter ein PDA. Alle wollen immer über die aktuellen Termindaten auf dem Laufenden sein. Daher werden die Termindaten über einen zentralen Server verwaltet, auf dem CalDAV arbeitet. CalDAV beinhaltet im Wesentlichen drei Funktionalitäten. Man möchte neue Termindaten auf dem Server ablegen können. Hierzu wurde eine Methode mit dem Namen MKCALENDAR implementiert. Des Weiteren muss es eine Möglichkeit geben, die Kalenderinformationen auf dem CalDAV Server abzurufen. Dabei besteht auch die Möglichkeit, die Termindaten einer anderen Person zu durchforsten. Jedoch möchte das vielleicht nicht jede Person. Das bringt uns zur Sicherheitsfunktionalität. Hierüber kann man einstellen, wer auf die

```
>>Anforderung<<
GET /bernhard/calendar/inbox/mtg456.ics HTTP/1.1
Host: cal.example.com

>>Antwort<<
Date: Thu, 02 Sep 2004 17:05:23 GMT
Content-Type: text/calendar
Content-Length: xxxx

BEGIN:VACALENDAR
  VERSION:2.0
  PRODID: -//Example Corp.//CalDAV Server/EN
  BEGIN:VEVENT
    DTSTAMP:20040901T200200Z
    DTSTART:20040902T130000Z
    DTEND:20040902T140000Z
    SUMMARY:CalDAV draft review
    UID:34222-232@example.com
  END:VEVENT
END:VACALENDAR
```

Termindaten zugreifen darf und wer nicht. Dabei können auch Schreibrechte vergeben werden. Abschließend schauen wir uns noch ein praktisches Beispiel an. Die Webseiten Anforderung beim HTTP Protokoll erfolgt über die GET Methode. Auch die Kalenderdaten werden über die GET Methode angefordert. Wir fordern den Inhalt der Datei mtg456.ics an. Als Antwort wird uns der Inhalt der Datei zurück geliefert. Da wir das iCalendar Format ausführlich diskutiert haben, soll auf die zurück gelieferten iCalendar Daten nicht weiter eingegangen werden. Alle Kommunikationsvorgänge zwischen Server und Client finden nach diesem Prinzip statt. [IEEECAL]

## 5.4 Diskussion: Vergleich der Technologien

Ein Vergleich dieser Technologien ist schwer, da jede Technologie ihr eigenes Anwendungsgebiet hat. Dennoch soll an dieser Stelle eine kurze Diskussion versucht werden. Zuerst widmen wir uns dem iCalendar Format.

Es ist zwar richtig, dass iCalendar ein universelles Format zur Speicherung von Kalenderdaten ist, aber was bring solch ein Format, wenn es nicht in jeder Kalenderapplikation implementiert ist? Zwar steigt die Zahl der Anwendungen, die das Format unterstützen täglich, jedoch gibt es noch immer viele Anwendungen, die diese Funktionalität nicht bieten. Beispielsweise bieten beide webbasierten Kalendersysteme, die wir bisher kennen gelernt haben, diesen Standard nicht an. Jedoch gibt es noch weitere Ungereimtheiten: Wir haben gesehen, das iCalendar sogar Journals unterstützt. Jedoch verfügt nicht jedes Kalendersystem über die Möglichkeit, den Tagesablauf zu protokollieren. Was also tun, wenn eine Anwendung eine iCalendar Datei mit solchen Einträgen exportiert und die Zielanwendung diese Journal-Einträge nicht versteht? Es gibt zwei Möglichkeiten: Entweder verweigert die Anwendung den Import der kompletten iCalendar Datei, oder die Anwendung importiert nur die Passagen der Datei, die sie versteht und ignoriert den Rest. An dieser Stelle gibt es Verbesserungsmöglichkeiten. Jedoch werden diese nur Erfolg haben, wenn die Hersteller einschlägiger Terminsoftware mitziehen.

Abschließend soll noch ein Wort zu den Synchronisationsstandards gesagt werden. Die Frage, ob CalDAV oder SyncML die bessere Lösung ist, lässt sich nicht global beantworten. Diese Entscheidung ist in erster Linie situationsabhängig. SyncML hat seine Stärken im Bereich der Synchronisation. Es ermöglicht problemlos, jeden beliebigen Datentyp zu synchronisieren. Wenn noch keine Geräte vorhanden sind und ein Synchronisationsserver geplant ist, ist SyncML eine gute Wahl. Wenn man womöglich im Unternehmen schon einen fertigen WebDAV Server bereit stehen hat oder über einige

Webserver verfügt, kann CalDAV die günstigere Alternative sein. Wenn man einen WebDAV Server besitzt, muss man ihn lediglich um die CalDAV Funktionalität erweitern. Unternehmen sind immer um eine kostengünstige Lösung bemüht. Je nachdem welche Hardware schon vorhanden ist, eignet sich einer der Standards. Zudem muss man sich auch die Frage stellen, ob man den Schwerpunkt auf Synchronisation von Endgeräten legen will oder nicht. WebDAV geht in seiner Funktionalität beispielsweise weit über den Synchronisationsprozess hinaus.

Zusätzlich stellt sich die Frage wie viele Endgeräte über solch einen Server verwaltet werden. Wenn es eine geringe Zahl ist, benötigt man nicht so viel Rechenleistung. Womöglich reicht es dann aus, einen bereits vorhanden Rechner um solch eine Serversoftware zu erweitern. Wenn sehr viele Geräte ins Spiel kommen, sollte man sich auch Gedanken zu den Verbindungskosten machen. Gerade Handys und PDAs nutzen meistens das Mobilfunknetz als Übertragungsweg. Wenn viele Übertragungen statt finden, verursacht dies nicht zu vernachlässigende Kosten. Das Unternehmen muss also eine Kosten-Nutzen Rechnung aufstellen. In diesem Zusammenhang wären die folgenden Fragen zu beantworten: Welchen Vorteil bringt mir diese Technologie? Ist das mit dieser Technologie gelöste Problem ein zentrales Problem des Unternehmens? Welche Kosten entstehen?

Natürlich kommen diese Technologien auch beim Heimanwender zum Einsatz. Jedoch handelt es sich hierbei nicht um das Einsatzfeld eines Unternehmens. Der Heimanwender hat ganz andere Ansprüche. Ihm genügt es meistens, am Abend seine Daten mit dem PC abzugleichen und muss nicht jede kleine Änderung sofort an einen globalen Server senden. Auch ist es unwahrscheinlich, dass er Zuhause eine Sekretärin hat, die für ihn die Termine macht. Daher haben Synchronisationstools, die im Heimbereich zum Einsatz kommen, solche Funktionalitäten, wie die Synchronisation in Echtzeit, meist erst gar nicht implementiert.

Synchronisationsstandards entstanden erst in der Neuzeit der Informatik. Daher ist abzuwarten, wie sie sich weiter entwickeln. Da der Trend aber immer weiter zu vielen verschiedenen Multimediageräten geht, wird dieses Thema in der Zukunft noch wichtiger werden. In den vorgestellten Synchronisationsstandards steckt das Potential, diese Aufgabe zu meistern. Man darf gespannt sein, wie sich diese in Zukunft weiterentwickeln werden.

# 6 Das Webkalendersystem Cross-Calendar

Nachdem wir einen groben Überblick über gängige Kalendersysteme erlangt haben, wollen wir uns jetzt ganz dem Webkalendersystem Cross-Calendar widmen, an dem ich im Rahmen meiner Praxisphase mitgearbeitet habe. Der Cross-Calendar wurde als Komponente des Cross-Desktops entwickelt. Der Cross-Calendar ist somit keine Standalone Anwendung. Daher wollen wir zu Beginn dieses Kapitels eine kurze Einführung in den Cross-Desktop machen. Anschließend soll dann der Cross-Calendar und dessen Funktionalität vorgestellt werden. Im dritten und letzten Abschnitt wollen wir uns dann den Cross-Calendar aus technischer Sicht betrachten. Da das Thema der Abschlussarbeit die Analyse von Kalendersystemen ist, soll dabei nicht so sehr ins Detail gegangen werden. Auf Klassendiagramme wollen wir ganz verzichten. Es soll nur das prinzipielle Konzept dargestellt werden. Also wie die Kalenderdaten gespeichert werden und wie die Kommunikation zwischen Server und Client von statten geht. In diesem Zusammenhang soll erklärt werden, warum genau diese Vorgehensweise und keine andere Vorgehensweise angewendet wurde. Der Cross-Desktop verfolgt eine bestimmte Philosophie, auf die im nächsten Abschnitt näher eingegangen wird. Daher musste der Cross-Calendar, als Teil des Cross-Desktops, dieser Philosophie folgen. In Kapitel sieben wollen wir dann den Cross-Calendar mit den anderen am Markt befindlichen Kalendersystemen und Technologien vergleichen und die Vor- und Nachteile des Cross-Calendars herausarbeiten.

## 6.1 Die Webapplikation Cross-Desktop

Der Cross-Desktop ist eine Applikation, die Dateimanagement, Newsmanagement und E-Mail Management in nur einer Anwendung vereinigt. Die Applikation ist vollständig webbasiert. Mit einem Internetzugang und aktiviertem JavaScript kann man die Anwendung mit nahezu jedem Browser nutzen. Nachfolgend wollen wir uns die Funktionen dieser Webapplikation etwas näher anschauen. Um Missverständnissen vorzubeugen, folgender Hinweis: Damit die Screenshots nicht so groß werden, ist auf den folgenden Abbildungen nur der Cross-Desktop zu sehen. Auf das darum befindliche Fenster des Webbrowsers wurde verzichtet.

Auf der nächsten Seite ist das Hauptfenster des Cross-Desktops abgebildet. Auf den ersten Blick sieht die Webapplikation wie ein Dateiexplorer des lokalen Einzelplatzrechners aus. Dies vereinfacht das Einarbeiten in die Anwendung. Jeder hat schon einmal mit Dateiexplorern gearbeitet. Ein Großteil der Bedienung kann daher intuitiv erlernt werden. Oben sehen wir ein Menü. Durch Klick gelangt man zu weiteren Untermenüpunkten. Über das Menü kann man so die Ansicht ändern oder Operationen auf Dateien anwenden, wie zum Beispiel kopieren oder umbenennen. Die darunter befindliche Symbolleiste ermöglicht einen Schnellzugriff auf zentrale Funktionen des Cross-Desktops. Über das Menü sind auch die E-Mail und News Funktionalitäten zu erreichen. Dazu kommen wir jedoch später. Diese Elemente sind in der Abbildung mit dem Indice (1) gekennzeichnet. Vom Dateiexplorer her kennt man eine Adressleiste, über die man schnell in einen anderen Ordner wechseln kann, wenn man dessen Pfad auswendig weiß. Auch der Cross-Desktop bietet solch eine Möglichkeit über ein „Wechseln nach“ Eingabefeld an (2). Darunter (3) wird dem Benutzer gezeigt, in welchem Verzeichnis er sich gerade befindet. Des Weiteren gibt es die Möglichkeit, den zu

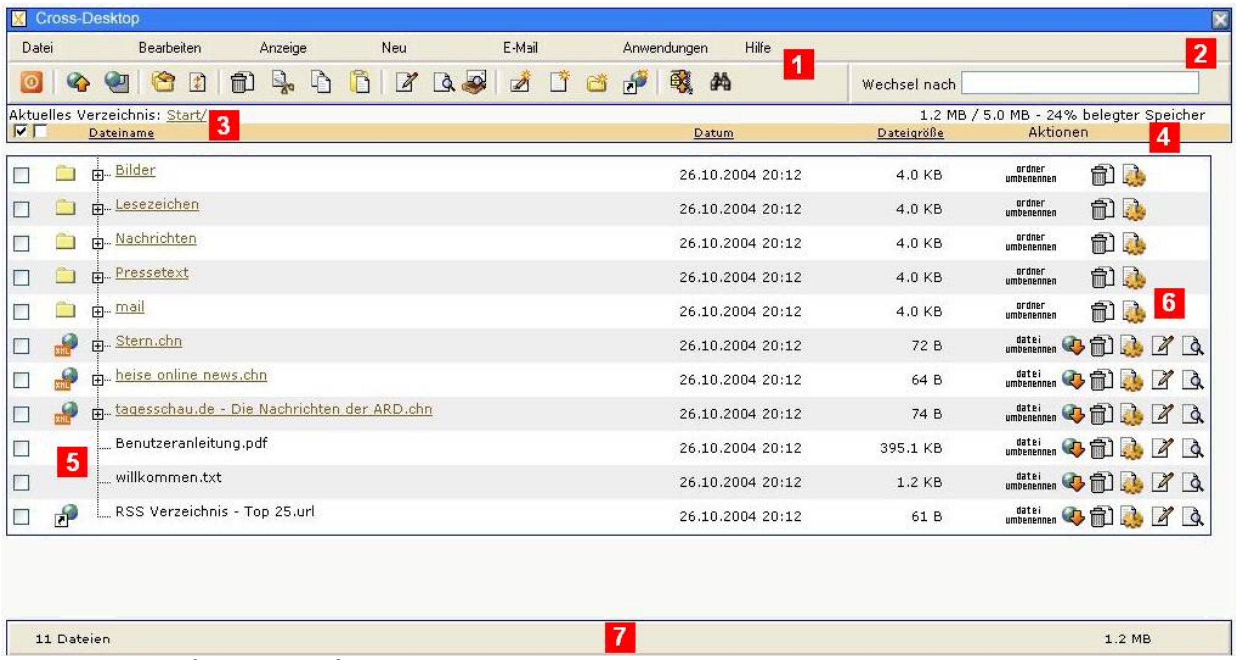



*Abb. 44 - Hauptfenster des Cross-Desktops*

verwendeten Speicherplatz auf dem Webserver einzuschränken. Rechts daneben (4) hat der Benutzer daher immer den Überblick darüber, wie viel freier Speicherplatz ihm noch zur Verfügung steht. Über die Überschriften Dateiname, Datum, ..., (3) kann man die Dateien des Verzeichnisses sortieren. Eine Sortierung nach der Dateigröße ist beispielsweise möglich. Dabei hat man noch die Auswahl zwischen einer aufsteigenden und einer absteigenden Sortierung. In jedem Fenster erwartet man ganz unten eine Statusleiste. Auf diese wurde auch im Cross-Desktop nicht verzichtet (7). Sie zeigt immer die Anzahl der Dateien und Ordner des aktuellen Verzeichnisses an, sowie deren gesamten Speicherplatzverbrauch. Der Inhalt des Ordners wird in der Mitte des Cross-Desktops angezeigt (5). Wir sehen Unterordner und Dateien. Manche dieser Dateien sind auch Bookmarks. In unserer Abbildung sind die Bookmarks importierte RSS-Feeds. Dazu kommen wir noch später. Jede Datei oder jedes Verzeichnis ist mit einem Satz von Symbolen versehen (6). Über diese Symbole kann man die Datei umbenennen, herunterladen, Dateirechte vergeben, bearbeiten und löschen. Wenn man auf mehrere Dateien die gleiche Operation anwenden möchte, kann man die gewünschten Dateien über die Checkboxen vor den Dateien (5) auswählen und dann die Dateioperation über das Menü oder die Symbolleiste aktivieren. Die Verzeichnisansicht bietet noch weitere Informationen, wie Dateigröße und Erstellungsdatum.

Wie eingangs schon erwähnt, beinhaltet der Cross-Desktop drei wesentliche Funktionen: Dateimanagement, Newsmanagement und E-Mail Management. Im Nachfolgenden sollen diese drei Komponenten vorgestellt werden:

Vom Dateimanagement haben wir schon einiges in der Einführung in die Oberfläche des Cross-Desktops gehört. Daher sollen hier nur noch ein paar Ergänzungen folgen. Der Cross-Desktop unterstützt, wie der herkömmliche Dateiexplorer auch, eine Multiwindowfunktionalität. Das heißt, dass man Verzeichnisse in einem neuen Fenster öffnen kann, anstatt im aktuellen Fenster. So kann man mit mehreren Fenstern gleichzeitig durch seine Dateien navigieren. Auch ist erwähnenswert, dass der Cross-Desktop die Zip-Funktionalität beherrscht. Er kann ausgewählte Dateien zu einer Datei komprimieren oder ein vorhandenes Archiv entpacken. Die eigenen Verzeichnisse können

durchsucht werden. Dabei ist sogar eine Volltextsuche möglich. Über Datei->Hochladen können neue Dateien vom lokalen Rechner in das Cross-Desktop System übertragen werden. Derzeit kann leider immer nur eine Datei gleichzeitig hochgeladen werden. Das Bearbeiten von Dateien ist nur in begrenztem Maße möglich. Einfache Textdateien können bearbeitet werden, wie die rechts stehende Abbildung zeigt. Unter Textdateien fällt beispielsweise auch der Quelltext von Programmiersprachen. Andere Dateiformate können derzeit nicht bearbeitet werden. Auch die Anzeigefunktionalität von Dateien hat ihre Grenzen. Textdateien und Bilder können problemlos dargestellt werden. Welche Dateien dargestellt werden können, hängt von den Fähigkeiten des Browsers ab, in dem der Cross-Desktop ausgeführt wird.

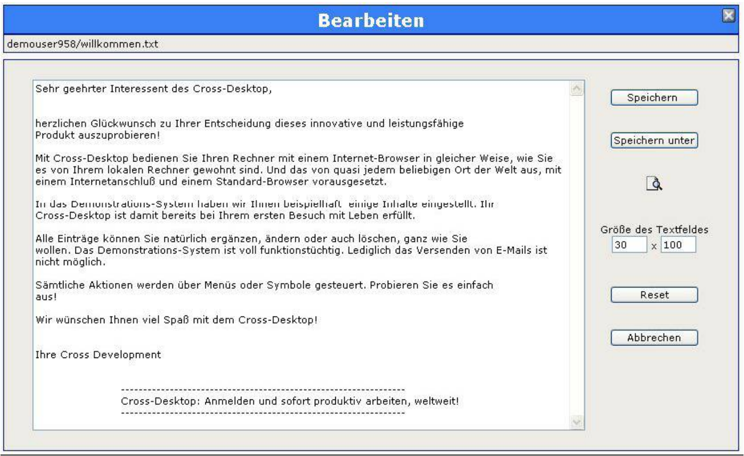


Abb. 45 - Bearbeiten Dialog des Cross-Desktops

Als nächstes wollen wir uns das E-Mail Management des Cross-Desktops anschauen. Allgemein kann man sich dieses Feature wie bei Outlook vorstellen. Man kann mehrere E-Mail Konten verwalten, deren Mails abholen und auch selbst Mails verfassen. Der einzige Unterschied besteht darin, dass diese Art der Mailverwaltung komplett webbasiert erfolgt. Die rechts stehende Abbildung zeigt die Verwaltung der E-Mail Postfächer. Es sind beliebig viele E-Mail Konten verwaltbar. Zur besseren Übersicht lassen sich die E-Mails der verschiedenen Konten farblich markieren. Es werden sogar verschiedene Methoden der Authentifizierung angeboten. Darunter zum Beispiel die POP Authentifizierungsmethode. Viele E-Mail Viren greifen den lokalen Rechner an. Da die komplette E-Mail Verwaltung online erfolgt, ist man daher vor Viren geschützt. Jede E-Mail wird als einzelne Datei gespeichert. Zur E-Mail Übersicht kommt daher der bereits vorgestellte Dateiexplorer des Cross-Desktops zum Einsatz. Man hat dann typische Ordner wie Posteingang und Postausgang zur Verfügung. Natürlich kann man auch individuelle Ordner anlegen. Die rechts stehende Abbildung zeigt die Ansicht einer E-Mail. Die Ansicht und die zur Verfügung stehenden Optionen sind typisch für ein E-Mail Programm. Daher soll an dieser Stelle nicht weiter darauf eingegangen werden.

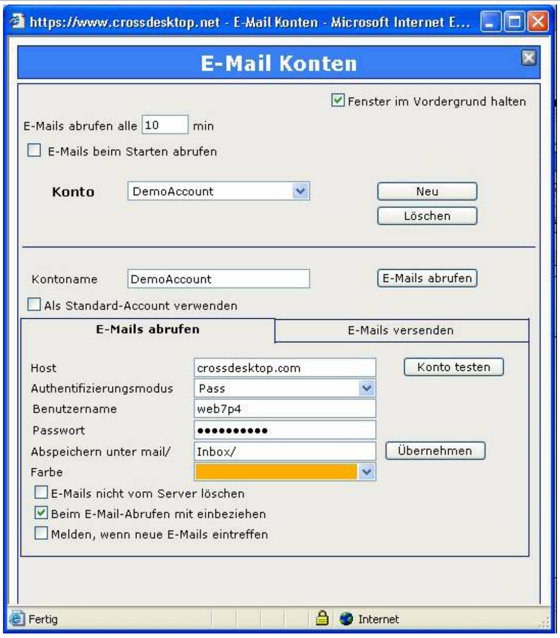


Abb. 46 - E-Mail Konten Verwaltung

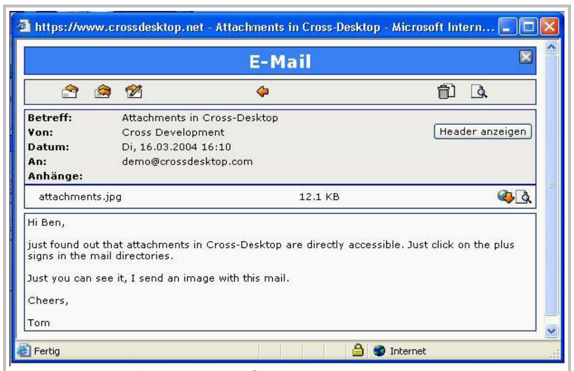


Abb. 47 - E-Mail im Cross-Desktop



*Abb. 48 - Optionsdialog für RSS*

Als letztes Feature nehmen wir uns das Newsmanagement System des Cross-Desktops vor. In diesem Zusammenhang sprechen wir von der Verwaltung von so genannten RSS-Feeds. RSS ist ein Internet-Standard für Nachrichten-Ticker. Mit den RSS-Feeds können aktuelle Informationen von Nachrichten-Anbietern aller Art abgerufen und im Cross-Desktop mit Titel und Link übersichtlich dargestellt werden. Die Darstellung der Links haben wir bereits in Abbildung 44 (5) gesehen. Rechts ist ein Optionsdialog abgebildet, über den man neue RSS Quellen in den Cross-Desktop eingeben kann. Anbieter solcher RSS Feeds sind beispielsweise Die Zeit, FAZ, Tagesschau, Heute, Spiegel, Stern und N24.

Des Weiteren beinhaltet der Cross-Desktop noch ein Adressbuch zur Verwaltung von Kontakten. Auf dieses Adressbuch kann man derzeit beim Schreiben von E-Mails zurück greifen. Der Cross-Desktop ist in Deutsch, Englisch und Französisch erhältlich.

Serverseitig benötigt diese Webapplikation einen Linux Server mit installiertem Apache und Perl. Perl bietet eine große Zahl von Modulen und ist für solch eine Applikation besonders geeignet. Zu diesem Thema wird in Kapitel sieben noch mehr gesagt werden. Clientseitig ist HTML, CSS und JavaScript erforderlich. Die Anwendung wird in zwei Versionen vertrieben: Einmal als Managed Service und einmal als Kaufversion. Der Managed Service kostet monatlich 4,50€. Hierbei erhält der Kunde eine installierte Version auf einem Webserver mit 500MB Speicherplatz. Die Kaufversion gliedert sich in zwei weitere Bereiche ein: Als Single-User und Multi-User Version. In der Single-User Version kann eine Person über die Anwendung online seine Daten verwalten. In der Multi-User Version können mehrere Benutzerkonten angelegt werden. Dies ist beispielsweise bei einem Unternehmen sinnvoll. Die Single-User Version ist derzeit für 49,90 € erhältlich und die Multi-User Version ist für 899€ zu haben. Bei der Kaufversion installiert der Kunde die Applikation auf seinem eigenen Webserver. Daher ist die Philosophie der Anwendung mit möglichst wenig Systemvoraussetzungen auszukommen. Ein Unix Webserver mit Perl (cgi) ist vollkommen ausreichend. Daher lässt sich die Anwendung auf vielen Webspaceangeboten installieren. Aufgrund der wenigen Anforderungen an den Server soll dem Benutzer die Installation so einfach wie möglich gemacht werden. Bei der Installation steht dem Benutzer zusätzlich ein in Perl geschriebener Installationsassistent zur Verfügung. Da keine Datenbank genutzt wird, werden alle Benutzerdaten in Dateien gespeichert. Auch clientseitig sollen so wenige Anforderungen wie möglich bestehen, damit der Cross-Desktop auf möglichst vielen Browsern und unter möglichst vielen Betriebssystemen genutzt werden kann. Die maximal verwaltbare Datenmenge ist auf die Ressourcen des Servers beschränkt. Wenn man nicht den ganzen Speicherplatz für den Cross-Desktop verwenden möchte, kann man, wie wir gesehen haben, eine Obergrenze eingeben. Dem Benutzer ist dann nicht gestattet, noch mehr Daten in den Cross-Desktop zu übertragen, obwohl eigentlich noch genug Serverressourcen vorhanden wären. Der Cross-Desktop ist als ein in sich geschlossenes System zu betrachten. Schnittstellen zur Außenwelt stellen die E-Mail Funktionalität, der Dateiupload und der Dateidownload dar. Der Cross-Desktop setzt verstärkt auf JavaScript, um möglichst viele Operationen ausführen zu können, ohne den Webserver zu belasten. Diese Philosophie wurde auch bei der Entwicklung des Cross-Calendars eingehalten.

## 6.2 Funktionsumfang von Cross-Calendar

In diesem Kapitel wollen wir uns die Funktionen der Webapplikation Cross-Calendar anschauen. Wir wollen an dieser Stelle mit der Termineingabe in den Cross-Calendar beginnen. Den hier eingegebenen Termin schauen wir uns anschließend aus technischer Sicht an. Abbildung 49 zeigt den Termindialog. Die einzelnen Attribute sind uns schon aus den vorigen Kapiteln bekannt. Als Beispiel soll wieder das Mittagessen in der Mensa genommen werden. Wir möchten von Montag, den 18.04.2005 bis Freitag, den 22.04.2005 jeweils von 12:00 bis 13:30 in die Mensa gehen. Daher legen wir einen entsprechenden Termin mit wöchentlicher Wiederholung an. An den Wochentagen Montag bis Freitag soll der Termin statt finden und an diesem besagten Freitag enden. Natürlich hätten wir auch einen täglichen periodischen Termin nehmen können. Die angehängte Datei erfüllt keine spezielle Funktion. Sie soll nur als Beispiel

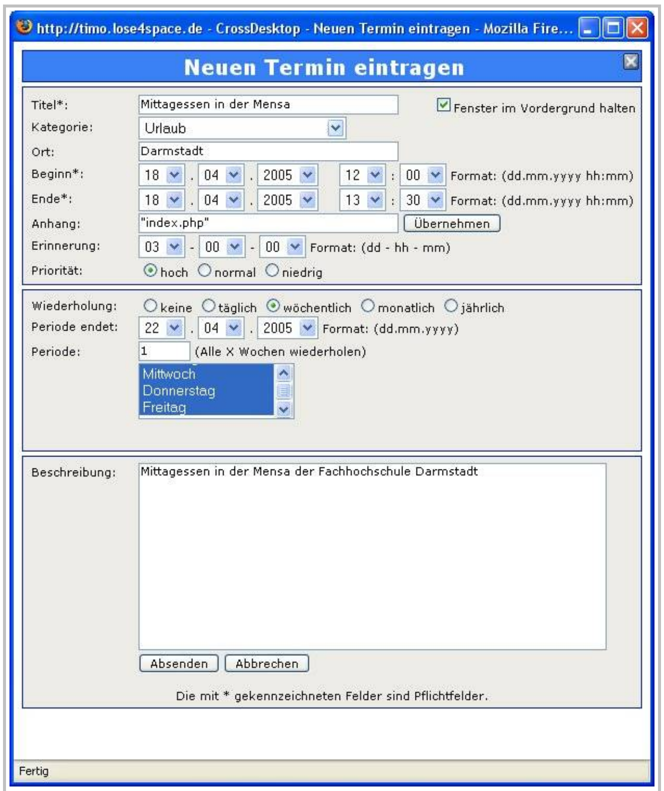



*Abb. 49 - Neuer Termin bei Cross-Calendar*

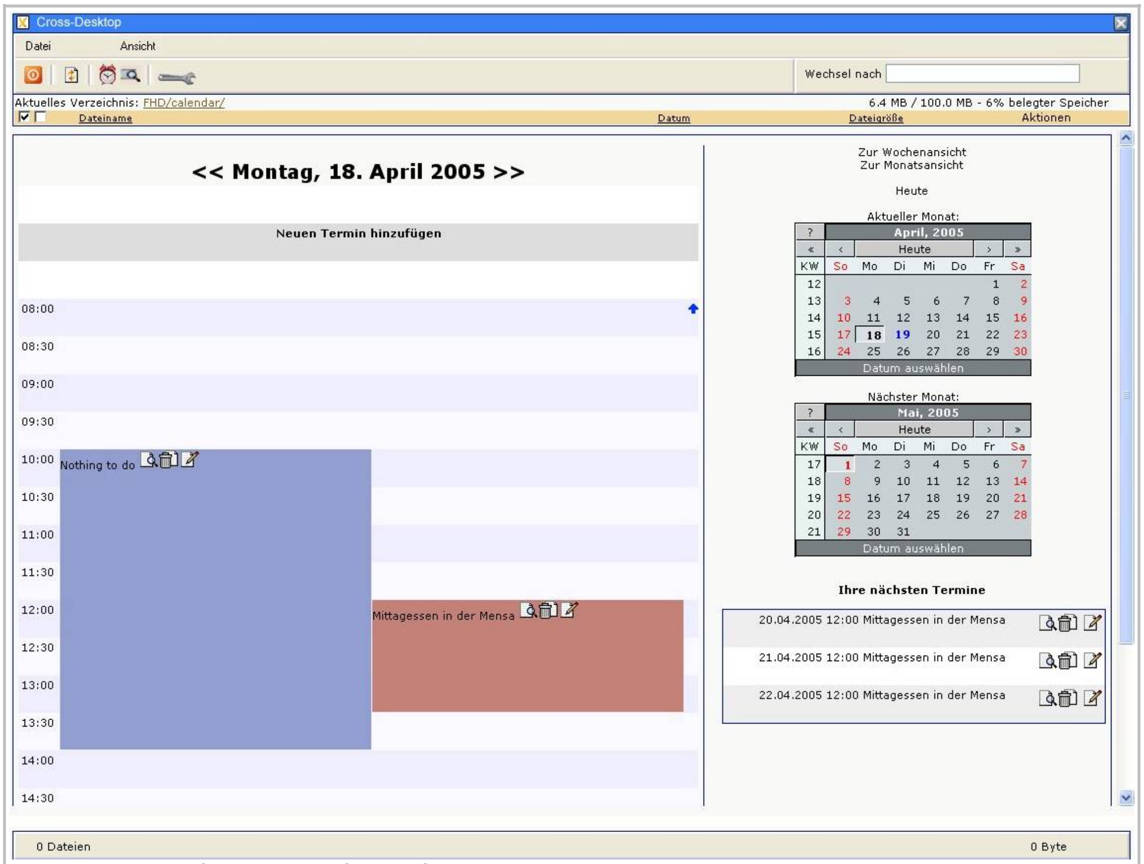



*Abb. 50 - Hauptfenster des Cross-Calendars*

für die spätere interne Betrachtung des Systems dienen. Dateien können im Cross-Desktop selektiert werden und durch Drücken des Übernehmen-Buttons ins Textfeld übertragen werden. Wir stufen den Termin als sehr wichtig ein und wollen drei Tage vor Terminbeginn erinnert werden. Wir weisen dem Termin die Kategorie Urlaub zu.

In Abbildung 50 ist die Oberfläche des Terminplaners abgebildet. Man erkennt auf den ersten Blick die Struktur des Cross-Desktops wieder. Über das Menü kann man zwischen den verschiedenen Terminansichten wechseln und neue Termine eintragen. Die wesentlichen Funktionen sind auch über die darunter befindliche Symbolleiste erreichbar. Wir können über die Symbolleiste das Fenster schließen oder bei Bedarf aktualisieren. Über den Wecker können wir neue Termine eintragen. Über die Lupe kommen wir zu einer Übersicht über die nächsten Termine. Das „Wechseln nach" Element wurde ebenfalls aus dem Dateiexplorer übernommen. Jedoch springt man bei einer Eingabe nicht zu einem Verzeichnis, sondern zum eingegebenen Datum. Rechts sehen wir zwei Kalender. Diese sind der OpenSource JSCalendar entnommen. Der erste Kalender zeigt den Monat des Tages, der gerade links in der Tagesansicht dargestellt wird. Darunter zeigt ein zweiter Kalender den darauf folgenden Monat an. Durch Klick auf einen Tag springt die Tagesansicht zu diesem Tag. Oben fallen noch die Symbole „<<" und „>>" auf. Hierüber kann man zum vorigen, beziehungsweise nächsten Tag weitergehen. Links unten befindet sich eine kleine Übersicht über die Termine, die als nächstes anstehen. Die Menge dieser Termine lässt sich über die Optionen einstellen. Diese lassen sich derzeit über den Schraubenschlüssel in der Symbolleiste erreichen. Die Tagesansicht ist typisch für ein Kalendersystem. Unser Mittagessen wird zwischen 12:00 und 13:30 dargestellt. Die Hintergrundfarbe gibt die Priorität des Termins wieder. Rot bedeutet hohe Priorität, blau bedeutet normale Priorität und grün bedeutet niedrige Priorität. In der Zeit von 10:00-14:00 findet auch noch der Termin „Nothing to do" statt. Dies ist ein einmaliger Termin, der außer Titel, Beginn und Ende keine weiteren Informationen beinhaltet. Auch diesen Termin werden wir uns im nächsten Kapitel anschauen. Da beide Termine teilweise parallel ablaufen, werden sie nebeneinander dargestellt. Die Breite der Termine ergibt sich dann aus der maximalen Anzahl von parallelen Terminen des jeweiligen Tages. Jedem Termin sind typische Cross-Desktop Symbole zugewiesen, um zur Detailansicht zu gelangen, den Termin zu löschen oder den Termin zu editieren.

Da wir gerade bei den Terminansichten sind, wollen wir in diesem Zusammenhang gleich die anderen Terminansichten des Cross-Calendars betrachten. Die Wochen- und Monatsansicht sind in Abbildung 51 dargestellt. Die Wochenansicht ist, anders als bei



*Abb. 51 - Wochenansicht und Monatsansicht in Cross-Calendar*

vielen Terminplanern, als Liste dargestellt. Zu jedem Tag werden alle Termine aufgelistet. Zu jedem Termin werden auch die Symbole, die uns schon aus der Tagesansicht bekannt sind, angezeigt. Tage, die einen Termin beinhalten, werden blau hinterlegt. Alle anderen Tage werden grau hinterlegt. Montags-Freitags sehen wir unseren Mensatermin. Am Montag kommt der „Nothing to do“ Termin dazu. Die Termine werden in der Wochenansicht bezüglich ihres Beginns sortiert. Der frühste Termin steht an erster Stelle. Die Monatsansicht verfolgt das gleiche Prinzip. Tage mit Termindaten werden blau hinterlegt. Wenn man mit der Maus über einen Tag mit Terminen geht, werden diese Termine in einer Tooltipp-Box mit Beginn und Titel aufgelistet. Ein direkter Zugriff auf den Termin ist in dieser Ansicht nicht möglich.

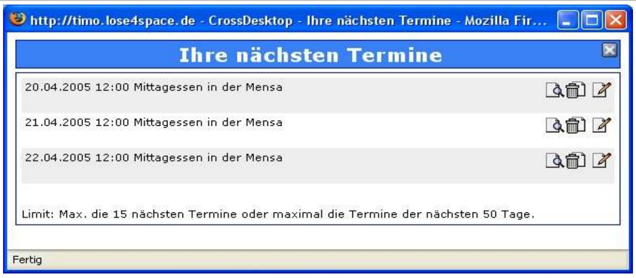



*Abb. 52 - Nächste Termine im Popup*

Zuvor haben wir gesehen, dass man unten links im Hauptfenster immer eine kleine Box zur Verfügung hat, die einem die nächsten Termine anzeigt. Dieses Feature gibt es auch als separates Fenster, wie die Abbildung 52 zeigt. Über die Optionen kann der Benutzer einstellen, wie viele Termine maximal dargestellt werden sollen. Des Weiteren kann er bestimmen, wie viele Tage er maximal in die Zukunft blicken möchte. Auch hier kann der Termin direkt angesehen, gelöscht oder editiert werden.




*Abb. 53 - Details eines Termins*

Abbildung 53 zeigt die detaillierte Terminansicht. Vom Layout her wurde die Ansicht von der E-Mail Ansicht des Cross-Desktops übernommen. Oben steht eine kleine Symbolleiste zur Verfügung, darunter werden die Termineigenschaften aufgelistet. Im unteren Bereich des Fensters wird dann die Beschreibung des Termins in einer transparenten Box mit blauem Rahmen dargestellt. Die Abbildung ist selbsterklärend. Daher soll an dieser Stelle nicht weiter darauf eingegangen werden.




*Abb. 54 - Terminerinnerung*

Zuletzt wollen wir uns die Terminerinnerung vornehmen. Diese erfolgt in Form eines Popups. Im Popup sieht man den Termin, an den gerade erinnert werden soll. Wenn mehrere Erinnerungen anstehen, sieht man eine Liste von Terminen. Jeden Termin kann man über die bereits bekannten Symbole betrachten. Außerdem hat man die Möglichkeit, sich nach ein paar Minuten, Stunden oder sogar Tagen erneut an diesen Termin erinnern zu lassen.

In der aktuellen Entwicklungsversion beinhaltet der Cross-Calendar keine Import- und Exportmöglichkeiten von Termindaten. Die Termindaten werden intern in einem nicht standardisierten Format gespeichert. Die Kalenderapplikation unterstützt keine Aufgaben. Im nächsten Abschnitt wollen wir uns die verwendeten Technologien, die bei der Entwicklung des Cross-Calendars zum Einsatz kamen, anschauen.

## 6.3 Cross-Calendar aus technischer Sicht

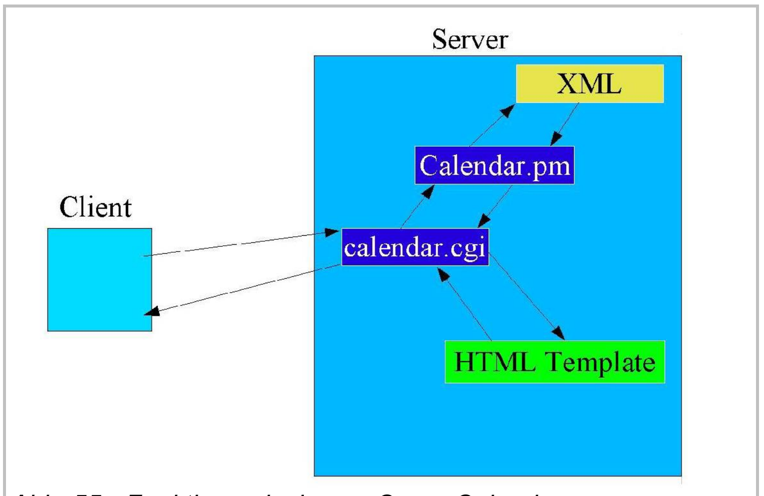



*Abb. 55 - Funktionsprinzip von Cross-Calendar*

In diesem Kapitel wollen wir einen groben Überblick über die technische Seite der Applikation bekommen. Wie alle anderen webbasierten Anwendungen beruht auch der Cross-Calendar auf dem Server-Client Prinzip, das uns schon aus Kapitel vier bekannt ist. Die Implementierungssprachen sind identisch mit denen des Cross-Desktops. Abbildung 55 zeigt eine grobe Skizze der System-Komponenten. Cross-Calendar arbeitet, wie der Cross-Desktop auch, mit so genannten HTML-Templates. Somit ist das Layout von der Programmierung vollständig getrennt. Bei einer Anfrage des Clients bindet das Perl Script dann einfach ein HTML-Template zum Script hinzu, ersetzt die Platzhalter des Templates durch konkreten Inhalt und schickt das Ergebnis zurück an den Client. Schauen wir uns beispielsweise Abbildung 50 an. Oben steht „Montag, 18. April 2005“. Im HTML-Template steht an dieser Stelle nur {date}. Das Script ersetzt dann zu Laufzeit diesen Platzhalter durch einen konkreten Wert. In diesem Fall eben den aktuellen Tag, den der Benutzer gerade anschaut. Dieses Verfahren hat im wesentlichen zwei große Vorteile: Man kann das Layout ändern, ohne etwas von der Implementierungssprache verstehen zu müssen. Auf diese Art ist es einfach, verschiedene Sprachen zu implementieren. Auch der Cross-Calendar unterstützt weitgehend die drei Sprachen des Cross-Desktops. Um dies zu erreichen bindet man neben dem HTML-Template noch eine Sprachdatei hinzu. Die in die HTML-Templates eingebundenen Inhalte sind im wesentlichen die Termindaten. Diese werden im Cross-Calendar im XML Format abgelegt. Da der Zugriff auf XML in einem komplizierten Algorithmus realisiert ist und dieser auch in fast allen Scripts des Cross-Calendars benötigt wird, wurde er in ein Modul ausgelagert. Bei Cross-Calendar heißt dieses Modul Calendar.pm. Ein Modul in Perl ist in etwa vergleichbar mit einer Klasse in einer anderen Programmiersprache. Jedoch gibt es eine große Zahl von Besonderheiten, auf die hier nicht weiter eingegangen werden soll. Das Modul verwaltet also die Termindaten und ermöglicht den Perl Scripts des Cross-Calendars einen einfachen Zugriff auf die Termindaten. Das Modul stellt hierzu eine große Zahl von Methoden zur Verfügung. Hier ein paar Beispiele für Methoden des Moduls: „Gib mir alle Termine zwischen X und Y“, „Gib mir alle Daten zum Termin mit der ID X“, „Trage einen neuen Termin ein“, „Liegt eine Erinnerung vor?“, „Lösche den Termin mit der ID X“. Für jede Ansicht und jedes weitere Fenster gibt es in Cross-Calendar ein eigenes Perl Script. Calendar.cgi ist beispielsweise für die Darstellung der Tagesansicht zuständig. Unter Verwendung der zuvor aufgezählten Komponenten generiert das Script dann eine Website bestehend aus JavaScript, CSS und HTML und sendet diese an den Client zurück. Der Client visualisiert dann das Ergebnis und führt die JavaScript Operationen aus, die in der HTML Datei enthalten waren. Welche das sind, werden wir später klären.

Wir haben gesehen, dass die Termine in XML gespeichert werden. In Perl gibt es eine große Zahl von fertigen Modulen. Als primäre Anlaufstelle wäre http://www.cpan.org zu

nennen. Hier findet man auch Module, die das Verwalten von XML Dokumenten vereinfachen. Im Cross-Calendar wurde überwiegend XML::DOM verwendet. Vom Prinzip her funktioniert XML::DOM genau wie JavaScript DOM. Man kann einfach durch die Knoten (Tags) des Dokuments springen, neue Knoten einfügen, editieren oder löschen. Jedoch ist es sehr zeitaufwendig, eine große XML Datei einzulesen. Um dieses Problem zu umgehen, werden bei Cross-Calendar alle Termine in verschiedenen Dateien gespeichert. Daily.xml, Weekly.xml, Monthly.xml und Yearly.xml stehen für periodische Termine zur Verfügung. Für einmalige Termine steht für jeden Tag eine eigene XML Datei im Format JahrMonatTag.xml zur Verfügung. An dieser Stelle wollen wir uns das Speicherformat der Termine in XML etwas näher anschauen. Hierzu nehmen wir uns die beiden Termine des vorigen Kapitels vor.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<schedule>
  <appointment>
  <id>13</id>
  <child>0</child>
  <begin>1113825600</begin>
  <end>1113831000</end>
  <real_begin>1113825600</real_begin>
  <real_end>1113831000</real_end>
  <title>Mittagessen in der Mensa</title>
  <category>6</category>
  <location>Darmstadt</location>
  <priority>high</priority>
  <description>Mittagessen in der Mensa der
   Fachhochschule Darmstadt</description>
  <attachment>"index.php"</attachment>
  <reminder>4320</reminder>
  <end_period>1114214400</end_period>
  <period>1</period>
  <weekdays>1,2,3,4,5</weekdays>
  </appointment>
</schedule>
```

Rechts sehen wir unseren Mensatermin im XML Format. Der Wurzelknoten in jedem XML Dokument nennt sich schedule. Jeder Termin ist in einem appointment Tag und dessen Untertags gespeichert. In unserem Beispiel beinhaltet das Dokument nur einen Termin. Jedoch kann jedes XML Dokument beliebig viele Termine beinhalten. Jeder Termin besitzt eine id. Hierüber erhält der Termin einen eindeutigen Identifier und kann so beispielsweise beim löschen eindeutig identifiziert werden. Child gibt an, ob dieser Termin am nächsten Tag auch eingetragen ist. Wir haben gesehen, dass es für jeden Tag eine eigene XML Datei gibt. Somit ist es nicht verwunderlich, dass Termine über mehrere Tage auch in mehreren Dateien gespeichert werden. Das child Attribut ist beim löschen sehr hilfreich. So kann das Script sehen, ob es in der nächsten Datei ebenfalls den Termin löschen muss oder nicht. 1 bedeutet, dass es in der nächsten Datei, also am nächsten Tag, einen weiteren Eintrag gibt. 0 bedeutet, dass es keine weiteren Vorkommen dieses Termins in der folgenden Datei gibt. Alle Zeitangaben sind im UNIX Timestamp Format gespeichert. Der UNIX Timestamp zählt die Sekunden seit dem 1. Januar 1970 00:00 Uhr GMT. In begin und end wird der Beginn und das Ende des Termins bezüglich des aktuellen Tages gespeichert. In real_begin und real_end wird der tatsächliche zeitliche Rahmen des Termins gespeichert. Die Werte unterscheiden sich nur bei Terminen, die über mehrere Tage gehen. Beispielsweise geht ein Termin am aktuellen Tag von 0:00 Uhr bis 23:59. Tatsächlich geht der Termin jedoch vom Vortag um 12:00 bis zum übernächsten Tag um 14:00. Title speichert den Titel des Termins und über category wird die Kategoriezugehörigkeit des Termins festgelegt. Location beinhaltet den Ort des Ereignisses und über priority wird der Wichtigkeitsgrad des Termins bestimmt. Description speichert die genaue Terminbeschreibung. In unserem Beispiel haben wir dem Termin eine index.php angehängt. Diese finden wir im attachment Tag wieder. Reminder gibt die Zahl in Minuten an, zu der man sich vor Terminbeginn erinnern lassen möchte. Da es sich um einen wöchentlichen Termin handelt, wird dieser in der weekly.xml gespeichert. Endperiod gibt das Ende der Periode in Form eines Timestamps an und period legt fest, alle wie viel Wochen der Termin wiederholt werden soll. Weekdays zählt alle Wochentage

auf, an denen der Termin statt finden soll. Hier sind es die Zahlen eins bis fünf, was in der Realität den Tagen Montag bis Freitag entspricht. Wir erinnern uns, dass unser Mensatermin von Montag bis Freitag eingetragen werden sollte. Zuletzt werfen wir noch einen kleinen Blick auf einmalige Termine. Hierzu wollen wir uns den „Nothing to do“ Termin anschauen.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<schedule>
  <appointment>
  <id>14</id>
  <child>0</child>
  <begin>1113818400</begin>
  <end>1113832800</end>
  <real_begin>1113818400</real_begin>
  <real_end>1113832800</real_end>
  <title>Nothing to do</title>
  <category>1</category>
  <location />
  <priority>normal</priority>
  <description />
  <attachment />
  <reminder>0</reminder>
  </appointment>
</schedule>
```

Vom Aufbau her ähnelt der Termin ganz dem zuvor gesehenen periodischen Termin. Da es sich um einen einmaligen Termin handelt, findet man hier keine periodischen Angaben. Die einzelnen Attribute sind uns schon aus dem vorigen Beispiel bekannt. Wir erinnern uns, dass wir bei diesem Termin außer Titel, Beginn und Ende keinen weiteren Informationen angegeben haben. Standardmäßig wird, wie wir rechts sehen, jedem Termin die Priorität normal zugewiesen. Daher beinhalten viele Tags auch keinen Inhalt. Reminder steht auf 0. Dies bedeutet, dass keine Erinnerung erfolgen soll. Da der Termin am 18.04.2005 statt findet, wird er in der Datei 20050418.xml abgelegt.

Es stellt sich jetzt nur noch die Frage, wo diese Dateien gespeichert werden. Der Cross-Desktop verfügt über eine Multiuserverwaltung. Für jeden Benutzer wird ein Verzeichnis mit Konfigurationsdateien angelegt. Cross-Calendar legt die XML Kalenderinformationen in diesem Verzeichnis ab.

Nachfolgend soll dargestellt werden, wie die Kalenderdaten auf den Bildschirm des Benutzers kommen. Hier wollen wir uns auf das Beispiel der Tagesansicht beschränken. Bei HTML wurde das Konzept des Layoutens ohne Tabellen verwendet. Statt <table> Tags kamen ausschließlich <div> Tags zum Einsatz. Dies hat jede Menge Vorteile: Tabellen waren nie zum Layouten vorgesehen. Die modernen Browser der letzten Jahre unterstützen CSS so gut, dass sich der Umstieg lohnt. Stichwort "Aufgeblähter und komplizierter Quellcode“. Tabellen generieren komplizierte Strukturen. Div Tags bleiben übersichtlich. Auch ist die Dateigröße von diesen Div formatierten HTML Dokumenten kleiner. Es gibt immer mehr Ausgabemedien (Bildschirm, Drucker, PDA, Handy). Tabellenloses Layout lässt sich so besser an die Ausgabemedien anpassen. Des Weiteren lässt sich JavaScript DOM einfacher auf solche Strukturen anwenden, als auf Tabellen. In der Tagesansicht schreibt nicht Perl die Termine in das HTML-Template. Stattdessen generiert Perl nur JavaScript Aufrufe und sendet diese an den Client. Erst auf Clientseite werden diese Operationen ausgeführt, die die Termine in die Tagesansicht eintragen. Dieses Vorgehen verhindert eine zu große Beanspruchung des Servers. Schauen wir uns hierzu nochmals die Abbildung 50 an. Wir sehen, dass der Benutzer die Randzeiten des Tages ein- und ausklappen kann. Normalerweise müsste der Server dann eine neue Website generieren und diese an den Client zurück senden. Da jedoch JavaScript die Termine zeichnet, ist der Client in der Lage beim Aufklappen eines Zeitbereichs die Termine neu zu zeichnen, ohne den Server zu konsultieren. Auch bei der Terminerinnerung wird diese Philosophie verfolgt. Hier wird darauf verzichtet, jede Minute den Server zu fragen, ob neue Erinnerungen vorliegen. Stattdessen erhält ein JavaScript vom Server beim Start des Cross-Calendars alle relevanten Daten zur Terminerinnerung und läuft dann im Hintergrund. Die Erinnerungsverwaltung erfolgt in einem unsichtbaren

Frame. Eine Aktualisierung der Informationen muss nur selten erfolgen. Ein neuer Termin, ein geänderter Termin oder ein gelöschter Termin wären an dieser Stelle als Beispiele zu nennen. Auch an anderen Stellen des Cross-Calendars wurde nach Möglichkeit ein JavaScript entwickelt, um den Server zu entlasten. Da auch der Cross-Desktop diese Philosophie verfolgt und ohne JavaScript nicht funktionsfähig ist, ist dieses Vorgehen kaum verwunderlich. Zu erkennen ist, dass bei der Implementierung auf eine strikte Trennung von Layout und Scriptsprache geachtet wurde. Bei den Perl-Scripts wurde zusätzlich darauf geachtet, dass alle Operationen auf die XML Dokumente in einem Modul realisiert wurden. Die anderen Scripts verwalten die Termine über dieses Modul. An dieser Stelle sei nur angemerkt, dass auch die Terminerinnerung einen komplizierten Algorithmus beinhaltet. Wenn der Benutzer beispielsweise an einen Termin erinnert wurde, wird dies auch in einer Datei festgehalten. Somit ist gewährleistet, dass der Benutzer bei einem zweiten Besuch nicht die gleiche Erinnerung noch einmal erhält. Jedoch ist der Umfang dieses Abschnitts zu klein, um noch näher darauf eingehen zu können. Stattdessen wollen wir uns abschließend anschauen, warum genau diese Art der Implementierung verwendet wurde und keine andere.

In Kapitel 6.1 haben wir schon viel über den Cross-Desktop und seine Philosophie gelesen. Ein großer Teil der Entscheidung liegt eben genau in dieser Philosophie begründet. Serverseitig macht es wenig Sinn, den Cross-Calendar in einer anderen Sprache als Perl zu programmieren, da der Cross-Desktop auch in Perl realisiert ist. Zudem bietet Perl eine große Zahl an Modulen, die viele nützliche Funktionen zur Verfügung stellen. Die Stärke von Perl liegt bei Dateioperationen und Dateiverarbeitung. Auch ist Perl tief in UNIX integriert. So ist es auch möglich, über Perl UNIX Konsolenbefehle auszuführen. Da der Cross-Desktop unter Anderem ein Dateimanagementsystem ist, ist es nicht verwunderlich, dass Perl verwendet wurde. Perl ist eine sehr mächtige Programmiersprache und nicht nur auf den Einsatz im Internet beschränkt. PHP bietet beispielsweise bei weitem nicht diese Möglichkeiten. Wenn der Cross-Desktop viele Benutzer auf einem Server verwaltet, kommt der Aspekt der Serverbelastung zum Tragen. Daher war man bemüht, so viele Funktionen wie möglich in JavaScript und somit auf den Client auszulagern. Der Cross-Desktop vertritt die Philosophie ohne Datenbank auszukommen. Für eine Kalenderapplikation wäre eine Datenbank sehr sinnvoll. Aufgrund dieser Philosophie war eine Datenbank jedoch tabu. Kalenderdaten unformatiert abzulegen, hätte einen umfangreichen Algorithmus nach sich gezogen. Da Perl XML unterstützt, fiel die Wahl auf XML, um den Algorithmus noch einigermaßen überschaubar zu halten. Zugegebenermaßen wäre es sinnvoller gewesen, die Kalenderdaten im iCalendar Format zu speichern. Jedoch war uns dieses Format zum Zeitpunkt der Implementierung weniger bekannt. Zur Diskussion kommen wir jedoch im nächsten Kapitel. Das Konzept der HTML-Templates bestand schon im Cross-Desktop und wurde in den Cross-Calendar übernommen. Das Gleiche gilt für die Unterstützung von verschiedenen Sprachen. Der Cross-Desktop wurde größtenteils mit Hilfe von Tabellen layoutet. Das Konzept des Layoutens ohne Tabellen war ein Pilotprojekt im Bereich des Cross-Desktops. Es gab einige Schwierigkeiten und Probleme bei der Entwicklung. Jeder Browser interpretierte den Quellcode etwas anders. Wie die Screenshots zeigen, hat man das Problem jedoch einigermaßen in den Griff bekommen. In der aktuellen Version gibt es nur noch kleine Darstellungsprobleme.

Im nächsten und letzten Kapitel wollen wir diese Art der Implementierung kritisch betrachten. Zuvor haben wir vieles über Kalendersysteme gelesen. Dieses Wissen wollen wir jetzt mit dem Cross-Calendar in Verbindung bringen und diskutieren, was man verbessern könnte.

# 7 Diskussion über Cross-Calendar

Wir haben schon viel über den Markt der Kalendersysteme gehört. Im vergangen Kapitel haben wir auch viel über die Webapplikation Cross-Calendar gelernt, die ich im Rahmen meiner Praxisphase mitentwickelt habe. Als Abschluss dieser Arbeit soll daher der Bogen zwischen dem Cross-Calendar und den anderen Kalendersystemen auf dem Markt gezogen werden. Dabei sollen die Vor- und Nachteile des Cross-Calendars gegenüber diesen anderen Terminplanern herausgearbeitet werden.

Beginnen wollen wir einen Vergleich auf Benutzerebene. Wir wollen uns ansehen, was die anderen Kalendersysteme bieten und was der Cross-Calendar bietet. Danach wollen wir diskutieren, welche Features noch fehlen. Wir werden sehen, dass ein paar Basisfunktionalitäten noch fehlen. Des Weiteren wollen wir zu unserem Anwendungsfalldiagramm aus Kapitel drei zurück kehren und uns anschauen, welche Anwendungsfälle der Cross-Calendar abdeckt und welche nicht. Als Abschluss dieses Abschnittes wollen wir diesen Vergleich dann noch auf die Kalendersystemplattformen ausweiten.

Danach wollen wir uns die Implementierung des Cross-Calendars anschauen und aufzeigen, wo man etwas hätte anders machen können. Wir wollen diskutieren, ob XML wirklich das geeignete Speicherformat für die Termine ist. In diesem Zusammenhang werden wir erkennen, dass es eine große Zahl von alternativen Speicherformaten gibt, die teilweise besser geeignet gewesen wären. Die am naheliegendsten Alternativen wollen wir heraus greifen und genauer diskutieren, welche Änderungen an Cross-Calendar nötig wären, um diese Alternativen zu realisieren. Wir werden erkennen, dass viele dieser Standards mit einem vertretbar hohen Aufwand nachträglich eingebaut werden könnten. Eigentlich hätte man erwartet, dass man die Anwendung komplett hätte umschreiben müssen, um die Termine anders zu speichern.

Als Beispiele wollen wir uns überlegen, was zu tun wäre, um eine Datenbank statt XML zu verwenden. Dabei soll aufgezeigt werden, ob eine Datenbank eine Vereinfachung der Programmierung und eine Performancesteigerung mit sich bringen würde. Da eine Datenbank entgegen der Philosophie des Cross-Desktops ist, wollen wir eine weitere Alternative ins Auge fassen: Es soll gezeigt werden, was zu tun wäre, um die Termine im standardisierten iCalendar Format zu speichern und welche Vorteile sich daraus ergeben würden.

Diese ganzen Aspekte bringen uns dann zu einem Fazit dieser Abschlussarbeit. Da das Fazit wichtig ist, wird ihm ein eigenes Kapitel acht zugeteilt. Wir wollen noch einmal auf den Cross-Calendar zurück blicken und aufzeigen, wo die wesentlichen Schwächen liegen und wo noch ein großes Potential zu finden ist. Das Fazit soll sich jedoch nicht nur auf den Cross-Calendar beziehen. Vielmehr wollen wir auch noch einen Blick auf den großen Markt der Kalendersysteme werfen und aufzeigen, in welchen Bereichen noch Verbesserungsmöglichkeiten zu finden sind. Wie wir wissen, kommen heute Kalendersysteme in Kombination zum Einsatz. Wir wollen versuchen, ein fiktives Kalendersystem zu entwerfen, dass in allen Situationen den Anforderungen des Benutzers gerecht werden kann. Ob uns dieser Versuch gelingt?

Beginnen wollen wir an dieser Stelle, wie versprochen, mit dem Vergleich von Cross-Calendar aus Benutzersicht.

## 7.1 Cross-Calendar im Vergleich zu anderen Terminplanersystemen

Beim ersten Blick auf Cross-Calendar fehlen dem Benutzer vier Möglichkeiten: Das Verwalten von Aufgaben, eine Synchronisationsmöglichkeit, das Sichern von Termindaten und das Group Scheduling. Alle anderen Anwendungsfälle, die wir in Kapitel drei herausgearbeitet haben, werden durch den Cross-Calendar unterstützt.




*Abb. 56 - Ihre nächsten Termine / Miniaturansicht*

Die Möglichkeit, Aufgaben einzugeben, wurde nicht vergessen. Jedoch hatte der zeitliche Rahmen der Praxisphase nicht dazu ausgereicht, dieses Feature zu implementieren. Ein Speicherformat für die Termine gibt es ja schon. Dieses Format könnte man auch zum Speichern von Aufgaben verwenden. Vielleicht könnte man jeder Aufgabe noch ein Attribut „todo“ spendieren. Somit könnte man einfach die Aufgaben von den Terminen unterscheiden. Dadurch könnte man sogar Termine und Aufgaben in den gleichen XML Dateien speichern und die schon vorhandenen Algorithmen für die Verwaltung der Aufgaben verwenden. Es müsste lediglich noch ein Interface zur Eingabe von Terminen und deren Darstellung realisiert werden. Wir erinnern uns an die Tages-, Wochen- und Monatsansicht des Cross-Desktops. In jeder Ansicht hat man einen kleinen Überblick über die kommenden Termine. Diese kleine Ansicht ist zur Erinnerung noch einmal in Abbildung 56 zu sehen. Man könnte diese Ansicht durch eine Ansicht von noch zu erledigenden Aufgaben ersetzen. Die Ansicht über die nächsten anstehenden Termine gibt es, wie wir wissen, auch in einem Popup. Daher wäre ein Ersetzen dieser Ansicht durch eine Aufgabenliste möglich, ohne Einbußen in der Übersichtlichkeit in Kauf zu nehmen.

Auch der Aspekt des Group-Schedulings wurde nicht vergessen. Aufgrund von Realisierungsschwierigkeiten wurde diese Komponente vorerst zurück gestellt. In Kapitel sechs haben wir den Cross-Desktop auch als E-Mail Managementsystem kennen gelernt. Leider bot der Cross-Desktop zur Zeit meiner Praxisphase keine klare Schnittstelle zu diesem System an, über das man hätte Mails versenden können. Der Aufwand für die Implementierung des Group Scheduling wäre daher zu groß gewesen. Wenn es solch eine Schnittstelle gegeben hätte, hätte man das Terminformular nur durch eine Eingabemöglichkeit von E-Mail Empfängern erweitern müssen. Des Weiteren hätte man noch einen etwas umfangreicheren Algorithmus benötigt, der eingehende Mails mit Termindaten erkennt und diese dann automatisch in den Terminplaner einträgt. Die Termine hätten in den Mails im iCalendar-Format versendet werden können, da dieses Format auch eine große Zahl anderer Kalenderapplikationen versteht.

Zum Zeitpunkt der Entwicklung wurde der Aspekt des Datenaustausches und der Synchronisation als nicht so wichtig angesehen. Wichtiger war die Realisierung der Basisfunktionalitäten eines typischen Kalendersystems. Wie wir in dieser Arbeit gesehen haben, ist dieser Aspekt jedoch mit einer der wichtigsten Aspekte in einem Kalendersystem. Dadurch erklären sich die fehlenden Sicherungs- und Synchronisationsmöglichkeiten. Perl unterstützt, wie wir wissen, das Packen von Dateien. Es wäre recht einfach, alle XML Kalenderdaten zu packen und dann an den Client zwecks Datensicherung zu schicken. Auch wäre es weniger aufwendig, diese gepackte Datei über ein Interface hochzuladen und danach zu entpacken. Dieses Vorgehen würde dann dem Konzept der Datenwiederherstellung folgen. Da es sich bei diesem XML-Format um eine eigene Kreation handelt, wären diese Sicherungsdaten dann nur im Cross-Calendar

verwendbar. Später werden wir uns die Frage stellen, ob XML überhaupt die richtige Wahl für die Kalenderdaten war. Somit stellt sich eventuell gar nicht die Frage, ob man die XML Daten sichern sollte, da ein von Grund auf anderes Konzept zu bevorzugen wäre. Daher wollen wir dieses Thema jetzt nicht weiter vertiefen. Halten wir einfach fest, dass es aus Benutzersicht keine Sicherungs- und Synchronisationsmöglichkeiten gibt.

Abschließend wollen wir noch den Bogen zu den Plattformen schlagen. Ohne Frage ist dieses Kalendersystem in die Kategorie der webbasierten Kalendersysteme einzuordnen. Streng genommen stellen webbasierte Kalendersysteme keine Plattform da, da sie auf verschiedenen Plattformen lauffähig sind. Der Cross-Calendar läuft auf allen Browsern, die eine aktuelle Version von JavaScript und CSS unterstützen. Dabei ist es egal, ob Betriebssysteme wie Linux oder Windows verwendet werden. Auf der neusten Generation von PDAs oder Smartphones mit Internetanbindung sollte der Cross-Calendar auch lauffähig sein. Ob das auf den doch recht kleinen Displays Sinn macht, ist die andere Frage. In erster Linie ist Cross-Calendar für den Einsatz auf Einzelplatzrechnern oder Laptops konzipiert worden. Es ergibt sich somit ein wesentlicher Vor- und Nachteil. Es ist sehr vorteilhaft, dass man bei der Terminverwaltung nicht auf einen speziellen Rechner oder Laptop angewiesen ist. Auch bei einem Freund oder Arbeitskollegen kann man sich mal eben schnell in den Cross-Calendar einloggen und seine Termine verwalten. Nachteilhaft ist in diesem Zusammenhang die Voraussetzung einer bestehenden Internetanbindung. Ohne den Zugriff auf das Internet ist man von seinen Kalenderdaten abgeschnitten. Das gleiche gilt bei einem Ausfall des Servers, auf dem der Cross-Calendar installiert ist. Um den Cross-Calendar betreiben zu können, braucht man neben einem Client auch einen Server. Zwar kosten Server und Internetzugang heute nicht mehr die Welt, aber eine Einzelperson wird sich solch eine Anschaffung sicherlich überlegen. Womöglich reicht es ihr ja, wenn sie nur abends ihre Termine verwaltet. Oder sie bevorzugt ein PDA, da sie viel mobil unterwegs ist und nicht immer über eine Internetanbindung verfügt. Zuletzt sei jedoch noch angemerkt, dass im Cross-Desktop neben dem Cross-Calendar auch ein mobiles Dateimanagement-, E-Mail Management- und Newsmanagementsystem zur Verfügung stehen. Dies könnte sich wiederum positiv auf eine Kauf- oder Nutzungsentscheidung auswirken.

## 7.2 Kritik an der Implementierung von Cross-Calendar

Nun wollen wir uns der Implementierung von Cross-Calendar zuwenden. Zwei Punkte stechen besonders ins Auge: Das Speicherformat der Termine und der Aspekt der Synchronisation. Vermutlich wäre es sinnvoller gewesen, die Termine in einer Datenbank zu speichern. Der Zugriff auf die Termine wäre einfacher und effektiver gewesen. Diesen Aspekt wollen wir im folgenden Abschnitt etwas näher betrachten. In dieser Abschlussarbeit haben wir auch gesehen, dass Synchronisation von Kalenderdaten sehr wichtig ist. Damit Cross-Calendar wenigstens in geringem Maße dieses Feature unterstützt, ist die Implementierung von iCalendar als Pflicht anzusehen. Dann stellt sich doch auch die berechtigte Frage, warum man dann nicht gleich auch intern die Termindaten im iCalendar-Format abspeichert. Mit diesem Vorgehen würde man viele Probleme auf einmal lösen. Wir würden die Philosophie des Cross-Desktops nicht verletzen, da er weiterhin ohne Datenbank auskommen würde. Des Weiteren hätten wir schon intern ein standardisiertes Format, das wir bei einem Synchronisationsprozess einfach auslesen und weitergeben können. Eine lästige Konvertierung zwischen internem Format und Synchronisationsstandard würde komplett wegfallen. Dieser Aspekt ist so

wichtig, dass in Abschnitt 7.2.2 geklärt werden soll, wie man iCalendar in den Cross-Calendar implementieren könnte. Der Aufwand wäre gar nicht mal so hoch, wie zunächst angenommen. Wir haben gesehen, dass alle Operationen auf XML Dokumente in ein Perl-Modul ausgelagert wurden. Die Struktur des Moduls könnte erhalten bleiben. Jedoch greifen die Methoden des Moduls nicht auf XML zu, sondern auf Daten im iCalendar-Format. Unter Perl gibt es auch für dieses Format Hilfsmodule. Dazu kommen wir jedoch im dafür vorgesehenen Abschnitt. Nicht zu vergessen ist die Datensicherungsmöglichkeit. Wenn wir die Termindaten im iCalendar Format auslagern würden, könnte eine andere Kalenderapplikation mit diesen Termindaten etwas anfangen. So könnte man auch manuell die Termindaten von einem Kalendersystem zu einem anderen Kalendersystem transferieren. Wir sehen, dass XML wohl doch nicht die erste Wahl in Sachen Kalenderdaten ist. Es ist auch anzumerken, dass es derzeit kein XML-Schema gibt, das die Tagstruktur der Dokumentinstanzen festlegt. Stattdessen hätte man eine Speicherung im iCalendar-Format versuchen sollen, da diese Art der Speicherung praktisch nur Vorteile mit sich bringt. An dieser Stelle wollen wir noch ein paar Schwachstellen des Cross-Calendars ansprechen und diskutieren, wie man diese beseitigen könnte.

Das Konzept der Programmierung, das vom Cross-Desktop vollständig übernommen wurde, ist eigentlich sehr gut. Wir erinnern uns, dass dieses Konzept in der Abbildung 55 grafisch dargestellt war. Es gibt nur einen kleinen Kritikpunkt an der Implementierung des Moduls zur Verwaltung der in XML gespeicherten Termine. Terminerinnerung und XML Dateizugriff wurde zusammen in einem Modul realisiert. Eigentlich wäre es sinnvoller gewesen, die Terminerinnerung in ein eigenes Modul auszulagern. Eine Trennung hätte die Implementierung eines anderen Speicherformats, wie zum Beispiel iCalendar, weiter vereinfacht.

Eigentlich vertritt man die allgemeine Philosophie, dass das Menü der Anwendung die komplette Funktionalität abbildet und zugänglich macht. Eine darunter befindliche Symbolleiste vereinfacht den Zugriff auf häufig genutzte Funktionalitäten. Derzeit kann man die Ansichten Tages-, Wochen- und Monatsansicht nur über das Menü erreichen und die Optionen des Kalenders kann man nur über die Symbolleiste aufrufen. Ein klares Defizit, das noch zu beseitigen wäre. Wer genau aufgepasst hat, würde jetzt bemängeln, dass man diese Ansichten auch durch Links im Content-Bereich des Cross-Calendars erreichen kann. Dies ist in Abbildung 50 oben rechts zu erkennen. Vorzuschlagen wäre also den Ansichtswechsel auch in die Symbolleiste zu übernehmen und auf die Links im Content-Bereich komplett zu verzichten.

Viele Leute vertreten die Meinung, dass man mit aktiviertem JavaScript seinen Rechner potentiellen Angreifern öffnet. Daher haben diese JavaScript immer deaktiviert. JavaScript wurde von Anfang an verstärkt in den Cross-Calendar implementiert. Daher ist dieser ohne JavaScript nicht funktionsfähig. Ein Verzicht auf JavaScript nicht denkbar, ohne die Anwendung komplett neu zu programmieren. Man kann also nicht der Philosophie nachkommen, keine systemkritischen Funktionen in JavaScript zu implementieren. Im Cross-Calendar ist JavaScript mehr als nur eine optische Aufwertung des Systems. Dies ist nicht weiter problematisch, da schon der Cross-Desktop ohne JavaScript nicht lauffähig war.

In der heutigen Zeit gibt es fast keinen Browser ohne Popupblocker mehr. Popups sind aber sozusagen ein Statussymbol des Cross-Desktops. Auch im Cross-Calendar erfolgen viele Operationen auf Termine in einem Popup. Hier wurde auf eine Alternative zum Popup verzichtet. Wer einen Popupblocker nutzt, kann den Cross-Calendar leider nicht nutzen. Dies gilt analog auch für den Cross-Desktop. Aktuelle Blocker ermöglichen die

Definition von Ausnahmen. So können dem Cross-Calendar Popups erlaubt werden und gleichzeitig kann anderen Seiten die Einblendung von Popups verboten werden. Eine solche Konfiguration durch den Benutzer ist sehr einfach und meistens schon mit wenigen Klicks erledigt. Daher ist eine solche Konfiguration dem Benutzer zuzumuten.

Wir haben auch gesehen, dass der Cross-Desktop drei verschiedene Sprachen unterstützt. Die Unterstützung dieses Features ist in den Cross-Calendar vollständig implementiert worden. Jedoch sind die Menüs und Texte nur teilweise in Englisch und noch gar nicht in Französisch übersetzt. Als Grund wäre hier auch Zeitmangel zu nennen. Da die Übersetzungsdateien in Textform vorliegen, könnte auch eine Person ohne Programmierkenntnisse diese Übersetzung vornehmen.

Wir haben gesehen, dass im Cross-Calendar das Konzept des Layoutens ohne Tabellen verfolgt wurde. Da dieses Konzept sehr neu ist, stellt jeder Browser den Quellcode etwas anders dar. Diese Problematik ist auch als Browserkrieg bekannt. Bei der Entwicklung des Layouts wurde auf eine konsistente Darstellung in den Browsern Mozilla und Internet-Explorer geachtet. In anderen Browsern gibt es teilweise noch Darstellungsprobleme. Sogar in den Browsern Mozilla und Internet-Explorer hat das Layout noch so genannte Kinderkrankheiten. Dies sind meistens kleine Darstellungsprobleme, die dem Benutzer in der Regel nicht gleich ins Auge fallen. Eine weitere Aufgabe würde also darin bestehen, das Layout weiter zu verbessern.

Die Optionen des Cross-Calendars sind auch nur angedacht. Derzeit kann man nur einstellen, wie viele Termine in der Terminvorschau maximal anzeigt werden sollen und wie viele Tage diese Vorschau maximal in die Zukunft gehen soll. Des Weiteren kann man bestimmen, ob die Nebenzeiten in der Tagesansicht standardmäßig ein- oder außgeklappt sein sollen. Hilfreich wäre auch eine Option, die die Neben- und Hauptzeit des Tages frei definierbar macht. Der Cross-Desktop soll zukünftig nicht nur auf den deutschsprachigen Raum beschränkt sein. In Amerika verwendet man ein anderes Datumsformat als in Deutschland. Hilfreich wäre daher auch eine Option, die die Darstellungs- und Eingabeform eines Datums änderbar macht. Des Weiteren könnte man dem Benutzer über die Optionen eine eigene Farbvergabe für die Prioritäten der Termine ermöglichen. Man sollte auch die Möglichkeit haben, eigene Terminkategorien zu erstellen. Der Aspekt der Terminkategorieverwaltung erscheint recht umfangreich. Daher wäre es sinnvoll, dieses Feature gleich als eigenständige Komponente und nicht als Teil der Optionen zu realisieren.

Beginn und Ende von Terminen werden derzeit über DropDowns eingegeben. Für einen unerfahrenen Benutzer ist dies sicherlich hilfreich. Für einen erfahrenen Benutzer stellt dies jedoch eine zeitliche Behinderung dar. Vorzuschlagen wäre eine Eingabemöglichkeit über DropDowns und Textfelder. Welche Art dann genutzt wird, sollte dem Benutzer überlassen werden. Dies wäre durch eine weitere Option problemlos realisierbar.

Zusammenfassend können wir also sagen, dass noch viel Feinarbeit an der Applikation zu erledigen ist. Auf den ersten Blick sieht die Anwendung optisch ansprechend und funktionstüchtig aus. Wenn man jedoch genauer hinsieht, erkennt man viele kleine Schwachstellen. Der Cross-Calendar beinhaltet sicherlich auch noch ein paar Fehler, da der zeitliche Rahmen der Praxisphase einen ausführlichen Test nicht zugelassen hat. Nach dieser ausführlichen Diskussion wollen wir uns auf den folgenden Seiten ganz auf das zentrale Problem des Speicherformats der Termine stürzen. Zunächst wollen wir grob andenken, wie eine Datenbank einzubinden wäre und welche Vor- und Nachteile diese mit sich bringen würde. Danach wollen wir noch auf iCalendar als Speicherformat zu sprechen kommen.

### 7.2.1 Datenbank statt XML

Zunächst sollten wir uns die Frage stellen, welche Art von Datenbank wir verwenden wollen. Diese Frage ist jedoch leicht und schnell beantwortet: Im Bereich des Webs ist eine MySQL-Datenbank sehr weit verbreitet. Auch fallen für die Nutzung dieser Datenbank keine weiteren Lizenzkosten an.

Abb. 57 - MySQL

Die Datenbankanbindung würde nur eine Änderung des Cross-Calendar Moduls mit sich bringen. Perl bietet standardmäßig viele Möglichkeiten des Datenbankzugriffs an. Als Sprache kommt, wie an vielen anderen Stellen auch, SQL zum Einsatz. SQL würde viele Algorithmen, die nötig waren, um die XML Dateien zu verwalten, erheblich vereinfachen.

Eigentlich bräuchte man dann nur eine Tabelle für die Termine, eine Tabelle für die Terminerinnerung und eine Tabelle für Aufgaben, sofern diese in den Cross-Calendar implementiert wären. Wenn man jeder Tabelle ein weiteres Attribut „user“ spendieren würde, könnten sogar die Termindaten aller Cross-Desktop Benutzer in einer Tabelle gespeichert werden. Wir haben gelesen, dass der Cross-Desktop für jeden Benutzer ein Verzeichnis mit Konfigurationsinformationen anlegt. Wenn man eine Datenbank verwenden würde, könnte man dann auch gleich diese Einstellungen in der Datenbank ablegen. Überlegenswert wäre in diesem Zusammenhang auch, alle E-Mails in der Datenbank zu speichern. Wenn man all dies realisieren würde, wäre es sinnvoll, für jeden Benutzer eine eigene Datenbank zu erstellen.

Eine Datenbank würde eine ganze Reihe von Vorteilen für den Cross-Desktop mit sich bringen: Wie wir gesehen haben, ist die Datenverwaltung mit einer Datenbank in der Regel einfacher, als die Verwaltung von Daten über eine Datei. Man erspart sich auf diese Weise also jede Menge komplizierte Algorithmen. Auch sind die Suchoperationen in den Datenbanken optimiert, das heißt performanter, als ein eigener Suchalgorithmus für Dateien.

Wir haben gelesen, dass der Kunde den Cross-Desktop auch als Lizenz erwerben kann. Bei dieser Möglichkeit installiert der Kunde die Applikation auf dem eigenen Webserver. Wenn der Cross-Desktop eine Datenbank verwenden würde, müsste der Kunde zumindest auch über Grundkenntnisse in diesem Bereich verfügen. Da dies entgegen der Philosophie des Cross-Desktops ist, würde eine Datenbank also in keinem Fall in Frage kommen.

Zudem lässt eine Datenbank auch noch jede Menge Fragen offen: Welche Attribute soll ein Termin haben und wie kann eine Sicherung der Termine erfolgen. Auf diese Fragen hätte man keine spontane und eindeutige Antwort. Wenn Termine in einem standardisierten Format, wie iCalendar importiert werden würden, müsste ein umfangreicher Algorithmus entwickelt werden, der die Termine in das interne Format umrechnet. Analog wäre es bei einem Terminexport. Natürlich könnte man in der Datenbank jeden Termin im iCalendar Format abspeichern und nur relevante Attribute, wie Beginn, Ende, Titel und ID in einem extra Attribut ablegen. Jedoch stellt sich dann die berechtigte Frage, warum man denn dann nicht gleich beim iCalendar Format bleibt und die Termine gleich in dieser Form in Dateien speichert. Da eine Datenbank nicht in Frage kommen würde, wollen wir diesen Denkansatz an dieser Stelle fallen lassen und uns gleich überlegen, wie wir den iCalendar Standard in den Cross-Desktop einbinden könnten.

### 7.2.2 Implementierung des iCalendar Standards

*Abb. 58 - CPAN (Zentrale Anlaufstelle für Perl-Module)*

Als Abschluss des siebten Kapitels wollen wir uns überlegen, wie wir das iCalendar-Format in den Cross-Calendar implementieren könnten. Die Termindaten sollen dabei weiterhin in Dateien gespeichert werden. Jedoch nicht wie bisher durch Zuhilfenahme von XML. Stattdessen sollen die Termine im iCalendar Format abgelegt werden. Wir erinnern uns, dass wir die Termindaten in einzelne Dateien aufgetrennt haben. Eine große Datei hätte zu Performanceeinbußen beim Einlesen und Schreiben geführt. Ob nun in XML oder im iCalendar-Format gespeichert wird, das eigentliche Problem der Performance bleibt gleich. Das in Kapitel 6.3 vorgestellte Konzept zur Trennung der Termindaten in einzelne Dateien ist sehr gut. Daher sollte an diesem Konzept nichts verändert werden. Da dieses Konzept den Großteil des Algorithmuses im Cross-Calendar Modul ausmacht, kann sogar der Quellcode des Moduls größtenteils unverändert bleiben. Im Modul selbst wird nicht direkt in die XML Datei geschrieben, sondern es wird das Perl Modul XML::DOM verwendet. Dieses Modul parst eine XML Datei und stellt sie dem Programmierer als komplexe Datenstruktur zur Verfügung. Des Weiteren stellt das Modul eine große Zahl von Methoden bereit, über die diese Datenstruktur bearbeitet werden kann. Als Beispiele wären editieren, einfügen und löschen von Elementen zu nennen. Man findet auch Methoden, die die Navigation durch die Datenstruktur vereinfachen. Erstaunlicherweise findet man hier auch Methoden, wie getElementByID, die uns aus JavaScript DOM bekannt sind. Bei genauerem Hinsehen sieht man eine große Ähnlichkeit zu JavaScript DOM. Mit einem einfachen Befehl kann man am Ende der Bearbeitung die komplexe Datenstruktur wieder zurück in die XML Datei schreiben. Der Programmierer braucht sich um den Schreibeprozess nicht weiter zu kümmern.

Auf den ersten Blick sieht die Aufgabe, einen Algorithmus zu entwickeln, der Daten im iCalendar Format speichert, sehr umfangreich aus. Wenn man das komplette iCalendar-Format in einem Algorithmus abbilden müsste, wäre dies auch der Fall. Aber warum das Rad neu erfinden, wenn es das Rad schon gibt. Zuvor wurde schon erwähnt, dass Perl eine große Zahl von Modulen beinhaltet. Wem diese Zahl von Modulen nicht genügt, der kann derzeit an die 8000 weitere Module über das Comprehensive Perl Archive Network, kurz CPAN, beziehen. CPAN ist eine freie Community, bei der jeder Programmierer die Möglichkeit hat, selbst entwickelte Module anderen Programmieren kostenlos zur Verfügung zu stellen. Dieses Verfahren mahnt aber auch zur Vorsicht: Da jeder diese Möglichkeit hat, weiß man nicht, wie fehlerfrei und zuverlässig diese Module sind. Bei all diesen Modulen findet man neben sehr guten Modulen auch Module, die man besser nicht verwenden sollte. Immerhin können die Module durch die Benutzer bewertet werden, was einen ersten Anhaltspunkt zur Qualität eines Moduls gibt. Über http://search.cpan.org kann dieses Archiv durchsucht werden. Bevor wir also anfangen, einen eigenen Algorithmus zur Speicherung zu entwerfen, suchen wir dort doch einfach einmal nach iCalendar. Aufgrund dieser großen Zahl von Modulen ist es wenig verwunderlich, dass uns immerhin 28 Module angeboten werden. Am geeignetsten erscheint das Modul iCal-Parser, das eine iCalendar Datei einlesen kann und dem Programmierer als Datenstruktur zur Verfügung stellt. Dieses Modul beinhaltet also eine ähnliche Funktionalität wie unser XML::DOM Modul. Somit kann auch die Algorithmus - Struktur des Cross-Calendar

Moduls auch gleich bleiben. Wir müssen nur das iCal Modul einbinden und verwenden. Alle Dateien würden dann eben die Endung ics tragen. Vielleicht ist das jetzt etwas zu einfach dargestellt. Es sollte an dieser Stelle nur gezeigt werden, dass es in Perl bereits Module gibt, die iCalendar Dateien verarbeiten können. Daher würde sich der Aufwand beim Umstieg auf das iCalendar Format noch in Grenzen halten.

Wenn man sich für diesen Umstieg entscheiden würde, sollte man das iCalendar Format jedoch noch einmal genau studieren und dann überlegen, welche Attribute man als Terminbeschreibung nimmt. Wir haben gesehen, dass iCalendar auch eine große Zahl von internen Attributen unterstützt. Das sind Attribute, die der Benutzer nicht direkt eingibt, sondern Attribute, die vom System erstellt und verwaltet werden. Als Beispiele wären das Erstellungsdatum und die Anzahl der Änderungen eines Termins zu nennen. Zumindest um diese beiden Attribute sollte Cross-Calendar noch erweitert werden. Über weitere Attribute kann man geteilter Meinung sein.

Schon in der Einleitung des Kapitels haben wir gesehen, das iCalendar nur Vorteile mit sich bringen würde und dass es keinen Grund dafür gibt, beim XML Speicherformat zu bleiben. Mit der Änderung des internen Speicherformats wäre eine iCalendar Schnittstelle zur Synchronisation und Datensicherung viel einfacher zu realisieren, als mit einem nicht standardisierten Speicherformat. Wir haben auch gesehen, dass WebDAV und andere Synchronisationsprotokolle auf das iCalendar Format zurück greifen. Eine Unterstützung für solch ein Protokoll könnte auf diese Weise auch viel einfacher implementiert werden. Dadurch könnte man auch eine hohe Kompatibilität zu anderen Terminplanersystemen erreichen.

Bei all der Europhie soll abschließend jedoch noch auf eine kleine Schwachstelle beim Auftrennen der Termindaten in verschiedene Dateien hingewiesen werden. Wir haben gesehen, dass es für jeden Tag eine Datei gibt. Bei der Darstellung der Termine eines Tages greift man also gezielt auf eine Datei zu. Wenn es jedoch einen Termin über mehrere Tage gibt und dieser nur in der Datei eingetragen ist, an dem er beginnt, würde er an den Folgetagen nicht dargestellt werden. Als Lösung wurde dieser mehrtägige Termin dann in jede Datei geschrieben, an dem er angezeigt werden soll. Zu kritisieren wäre, dass der Termin öfters als nur einmal in den Datensätzen vorhanden ist. Nehmen wir an, dass wir den mehrtägigen Termin nur in der Datei speichern, an dem er beginnt. So müsste man bei der Tagesansicht alle zeitlich davor liegenden Tage durchsuchen, um zu sehen, ob es einen Termin gibt, der bis zum Tag reicht, der gerade dargestellt werden soll. Nur wie viele Tage soll man zurück gehen? Gibt es ein Maximum, wie lang ein Termin sein kann? Prinzipiell gibt es kein Maximum. Also müssten wir alle Datensätze, die zeitlich vor dem Tag liegen, der dargestellt werden soll, durchsuchen. Dann könnten wir genauso gut alle Termine in einer Datei speichern. Natürlich kann man kritisieren, dass ein Eintrag eines Termins, der einen Monat dauert sehr zeitaufwendig wäre. Das ist auch richtig, da der Termin in mindestens 28 Dateien eingetragen werden muss. Jedoch ist die Zeit für den Lesezugriff dafür optimal. Man geht davon aus, dass ein Termin wesentlich öfters gelesen, als geschrieben wird. Daher hat man sich für diese Möglichkeit der Auftrennung in verschiedene Dateien entschieden.

Trotz dieser kleinen Schwachstelle, über die man noch weiter nachdenken könnte, sollte das Konzept der Dateiauftrennung beibehalten werden. Das XML-Format sollte man jedoch fallen lassen und statt dessen auf das viel besser geeignete iCalendar-Format umsteigen. Dieser Umstieg wäre mit einem überschaubaren Maß an Aufwand zu erledigen.

# 8 Fazit

In diesem Abschnitt soll noch einmal auf die Praxisphase und die neuen Erkenntnisse dieser Abschlussarbeit zurück geblickt werden. Beginnen wollen wir diesen Rückblick mit der Betrachtung des Cross-Calendars. Nachdem man das siebte Kapitel gelesen hat, könnte man sich die Frage stellen, was denn nun überhaupt gut am Cross-Calendar ist. Wir haben gesehen, dass die Anwendung noch einige Fehler beinhaltet. Auch ist nicht abzustreiten, dass bei der Entwicklung eines Speicherformats für die Termine so einiges nicht so „optimal“ gelaufen ist. Dennoch steckt im Cross-Calendar ein Potential, das kein anderes webbasiertes Kalendersystem bieten könnte. Die E-Mail und Dateimanagement Funktion des Cross-Desktops ermöglichen dem Cross-Calendar ganz neue Perspektiven. Dank der E-Mail Schnittstelle wäre ein Group Scheduling einfach zu realisieren. Somit käme das Terminplanersystem auch für den professionellen Geschäftsmann in Frage. Auch das Anhängen von Dateien an Termine ist dank des bereits vorhandenen Dateimanagements einfach zu realisieren. Wir haben gesehen, dass im Cross-Calendar auch die neusten Layouttechniken zum Einsatz kommen. Durch diese Techniken ergeben sich neue Layoutmöglichkeiten, die die Bedienfreundlichkeit der Anwendung weiter verbessern. Beispielsweise lassen sich Eingabefelder durch Zuhilfenahme von JavaScript dynamisch ein und ausblenden. Diese Technik wurde bei der Eingabe der periodischen Optionen eines Termins verwendet. Der Cross-Calendar könnte in Sachen Funktionsumfang und Bedienfreundlichkeit das Niveau eines einzelplatzrechnerbasierten Terminplanersystems erreichen. Man könnte sich in diesem Zusammenhang sogar überlegen, auch eine Journal-Verwaltung zu implementieren. Somit könnte der Benutzer auch sämtliche Protokolle und Mitschriften des Tagesablaufs im Cross-Calendar einfach und komfortabel niederschreiben. Kaum ein anderes webbasiertes Kalendersystem würde solch eine breite Funktionalität bieten. Natürlich wird man bei der Nutzung des Cross-Calendars nie auf einen Internetzugang verzichten können, aber für Menschen ohne festen Arbeitsplatz wäre solch ein System eine Alternative. Zugegebenerweise ist es noch ein weiter Weg bis zu dieser visionären Darstellung des Cross-Calendars. Halten wir aber einfach fest, dass die Anwendung aufgrund einer falschen Vorgehensweise keine Fehlentwicklung ist. Mit ein paar Änderungen und Erweiterungen könnte der Cross-Calendar einen eigenen Stil entwickeln in dem viele Potentiale stecken. Cross-Calendar und Cross-Desktop würden sich gegenseitig ergänzen und könnten so ein gut eingespieltes Team werden.

Im Bereich Basisfunktionalität eines Kalendersystems gibt es wenig zu sagen. Nahezu jedes Kalendersystem ermöglicht dem Benutzer die Eingabe jeder nur denkbaren Terminart. Heute besteht die Aufgabe bei der Weiterentwicklung von Terminplanern eher darin, die Bedienfreundlichkeit zu verbessern und neue Techniken des Datenaustausches zu schaffen. Dabei ist es besonders im geschäftlichen Bereich wichtig, immer aktuelle Termindaten zu haben. Wir haben Techniken, wie SyncML gesehen, die bereits versuchen, solchen Anforderungen gerecht zu werden. Dennoch ist das Stichwort Standardisierung in diesem Bereich ein Thema mit Defiziten. Wünschenswert wäre ein globaler Standard, der von allen Kalendersystemen uneingeschränkt unterstützt werden würde. Dieses Denken wird jedoch Wunschdenken bleiben. Am Markt gibt es viele Hersteller, die gnadenlos um Marktanteile kämpfen. Wir haben zwar gesehen, dass Standards, wie iCalendar auf eine breite Zustimmung stoßen. Jedoch hat jedes Kalendersystem seine eigene Vorstellung von Terminen und deren Attributen. Im Grundlagenkapitel haben wir gesehen, dass Termine aller Kalendersysteme gewisse

Attribute, wie Beginn, Ende und Titel besitzen. Jedoch gibt es auch Attribute, wie beispielsweise Zugriffsberechtigungen auf Termine. iCalendar unterstützt diese Funktionalität. Was ist aber, wenn ein Terminplanersystem eine iCalendar Datei mit Zugriffsbeschränkungen erstellt, die eine andere Applikation importiert, die keine Zugriffsverwaltung unterstützt? Wird der Termin dennoch importiert und wird er dabei womöglich einer unberechtigten Gruppe von Personen zugänglich gemacht? Was ist, wenn im iCalendar-Format ein Attribut gesetzt ist, dass die importierende Kalenderanwendung nicht versteht? Was ist zu tun? Den Termin nicht importieren? Den Termin dennoch importieren und einen Informationsverlust in Kauf nehmen? Dies sind alles Fragen, auf die iCalendar keine Antwort liefert. Zwar ist iCalendar mittlerweile in viele Terminplanersysteme integriert. Jedoch lässt der Aspekt der vollständigen Integrierung an so mancher Stelle noch zu wünschen übrig. Nicht zu vergessen ist, dass es heute immer noch Anwendungen gibt, die gar kein iCalendar-Format verstehen. Der webbasierte Terminplaner von Web.de wäre an dieser Stelle zu nennen. Man kann daraus also das Fazit ziehen, dass Standardisierung ja schön und gut ist. Wenn sie aber nur teilweise oder gar nicht in Terminplanersysteme integriert wird, bringt sie dem Endanwender nicht sehr viel. Ähnlich verhält es sich auch mit den Protokollen, die wir kennen gelernt haben. Auch SyncML und besonders CalDAV sind noch nicht sehr weit verbreitet, obwohl sie von der Idee her eine Bereicherung für die Welt des Kalendersystems darstellen würden. Eine wesentliche Aufgabe in der Zukunft des modernen Terminplanersystems besteht also darin, solche Standards zu fördern und zu verbreiten.

In diesem Zusammenhang wollen wir noch einmal zum Grund zurück kommen, weswegen die Synchronisation und der Datenaustausch zwischen Kalendersystemen so wichtig geworden ist. Der Benutzer hat den Wunsch, seine Termine möglichst einfach und übersichtlich zu verwalten. Dabei möchte er seine Termine in jeder Situation zur Hand haben. Mobilität und Übersichtlichkeit sind zwei Gegensätze, die sich trotz des gleichnamigen Sprichworts nicht anziehen. Dies haben wir in Kapitel vier gesehen. Mobilität geht immer zu Lasten der Übersichtlichkeit und umgekehrt. Kapitel vier erweckt den Eindruck, dass es für jede Situation eine eigene Plattform und somit ein eigenes Kalendersystem gibt. Dies war auch der Grund, weswegen die Synchronisation so wichtig wurde. Jetzt fragt man sich wo dieser Trend enden soll. Hat man bald hunderte verschiedene Plattformen? Man kann beruhigt sein. Es sind auch Gegenbewegungen zu erkennen, die versuchen Plattformen wieder zusammenzuführen. Auch das hat Kapitel vier implizit dargestellt. Wir haben gesehen, dass man versucht die Funktionalität eines PDAs und die Mobilität eines Handys ein einem Gerät zu vereinen. Wir erinnern uns in diesem Zusammenhang an die Smartphones. Leider kann auch dieser Trend nicht bis zur Vereinigung aller Geräte in einem Gerät fortgeführt werden. Mit der derzeit zur Verfügung stehenden Technologie ist es schlichtweg nicht möglich, Mobilität und Übersichtlichkeit in einem Gerät zu vereinen. Daher begnügt man sich derzeit mit dem Kompromiss, mehrere Geräte parallel zu benutzen. Eine weitere Forschungsmöglichkeit würde also auch darin bestehen, Geräte zu entwickeln, die man überall mit hin nehmen könnte. Diese Geräte müssten ein breites Funktionsspektrum bieten, dessen Bedienung einfach, komfortabel und übersichtlich ist. Solch ein Bestreben ist nicht nur wichtig für Terminplanersysteme, sondern eher ein generelles Streben bei der Entwicklung neuer Plattformen. An viele andere Applikationen hat der Benutzer die gleichzeitige Forderung der Übersichtlichkeit und der Mobilität. E-Mails, Adressen und persönliche Dateien wären an dieser Stelle als einschlägige Beispiele zu nennen. Wir sehen also, dass es uns trotz größter Mühe mit dem heutigen Stand der Technik nicht gelingen kann, eine Plattform für Terminplanersysteme zu schaffen, die allen Anforderungen des Benutzers gerecht

werden kann. Jedoch ist nicht auszuschließen, dass zukünftige neue Technologien dieses Problem nicht irgendwann lösen können.

Diese Abschlussarbeit sollte einen kleinen Einblick in die Welt des Terminplaners geben. Sie sollte zeigen, mit welchen Problemen man heute zu kämpfen hat und wo der Trend hin geht. Natürlich konnte im Rahmen dieser Arbeit nur ein kleiner Einblick gegeben werden, der sich auf die wesentlichen Konzepte eines Terminplaners und dessen Systemumgebung beschränken musste. Wir haben iCalendar ausführlich behandelt, da dieses Format auch als Verbesserungsvorschlag für den Cross-Calendar vorgeschlagen wurde. SyncML allein ist ein Thema, das eine ganze Bachelorarbeit füllen könnte. In Sachen SyncML und CalDAV musste sich die Betrachtung leider nur auf das wesentliche Konzept beschränken. Durch diese Arbeit haben wir etwas Problembewusstsein vermittelt bekommen und gesehen, dass ein Terminplaner viel mehr Aufgaben hat, als nur die in Kapitel zwei beschriebene Basisfunktionalität. Wir haben viele Problemlösungen kennen gelernt, jedoch auch viele Stellen gesehen, an denen es noch keine vernünftigen Lösungen gibt. Man hat den Eindruck, dass das computerbasierte Kalendersystem kein alter Schuh der vergangen Jahre ist, sondern ein Bereich ist, in dem noch ein großes Potential steckt, das nur noch erforscht und entwickelt werden muss. Somit ist anzunehmen, dass sich in diesem Bereich auch in den kommenden Jahren noch viel entwickeln und herausbilden wird. Man darf also gespannt sein.

# 9 Anhang

## 9.1 Literaturverzeichnis

Alle in dieser Bachelor-Arbeit verwendeten Namen und Markenzeichen werden ohne Rücksicht auf ihr Copyright verwendet. Literaturhinweise werden in eckigen Klammern dargestellt. Sofern die Literatur über das Internet bezogen wurde, wird auf die entsprechenden Quellen an gegebener Stelle hingewiesen.

[CC05] : Dirk Fokken und Roland Fries , Cross-Consulting, 2004, http://cross-consulting.com, Stand: 03.05.2005

[MH05] : Mailhilfe.de, MAILHILFE, 2005, http://www.mailhilfe.de/faq-10-Cross.html-Desktop, Stand: 03.05.2005

[SUN02] : The Mozilla Organization, Mozilla Sunbird, 1998–2005, http://www.mozilla.org/projects/calendar/sunbird.html, Stand: 28.02.2005

[OUT05] : Microsoft Corporation, Microsoft Outlook, 2005, http://office.microsoft.com/de-de/FX010857931031.aspx, Stand: 28.02.2005

[KOG05] : KDE e.V., Korganizer, 2005, http://korganizer.kde.org, Stand: 06.03.2005

[DEWEB] : web.de, Webbasierter Terminplaner, 2005, http://www.freemail.de, Stand: 07.03.2005

[WIKICSV] : Wikipedia, CSV Format, 2005, http://de.wikipedia.org/wiki/CSV, Stand: 07.03.2005

[FREESCH] : Freenet, Terminplaner von Freenet, 2005, http://www.freenet.de, Stand: 18.03.2005

[DUAU] : Karl Mall, Musikverein Durlach-Aue e.V., 2005, http://www.mv-aue.de, Stand: 13.03.2005

[HRPRESS] : Herr Press, Herr Press, 2004, http://www.elmstreet.de, Stand: 13.03.2005

[TVBROW] : Martin Oberhauser, Til Schneider, Bodo Tasche , TV-Browser, 2005, http://www.tvbrowser.org, Stand: 13.03.2005

[FHDOBS] : Fachhochschule Darmstadt, Onlinebelegsystem, 2005, http://www.fbi.fh-darmstadt.de/~belegen, Stand 16.03.2005

[WIKIAL] : Wikipedia, Arbeitsplatzrechner, 2005, http://de.wikipedia.org/wiki/Arbeitsplatzrechner, Stand: 18.03.2005

[WIKIPC] : Wikipedia, Personal Computer, 2005, http://de.wikipedia.org/wiki/Personal_Computer, Stand: 18.03.2005

[WIKIWS] : Wikipedia, Workstation, 2005, http://de.wikipedia.org/wiki/Workstation, Stand: 18.03.2005

[WIKINB] : Wikipedia, Notebook, 2005, http://de.wikipedia.org/wiki/Notebook, Stand: 18.03.2005

[WIKILAP] : Wikipedia, Laptop, 2005, http://de.wikipedia.org/wiki/Laptop, Stand: 18.03.2005

[DOOOUT] : Paul Tiede, Kommunikation - Der Schlüssel zum Erfolg?, 2004, http://www.dooyoo.de/email-fax-datenuebertragung/microsoft-outlook-2003/1012048,

Stand: 19.03.2005

[WIKISC] : Wikipedia, Client-Server-System, 2005, http://de.wikipedia.org/wiki/Client-Server-Prinzip, Stand: 21.03.2005

[WIKISER] : Wikipedia, Webserver, 2005, http://de.wikipedia.org/wiki/Webserver, Stand 21.03.2005

[WIKICL] : Wikipedia, Webbrowser, 2005, http://de.wikipedia.org/wiki/Webbrowser, Stand: 21.03.2005

[WIKIHTTP] : Wikipedia, Hypertext Transfer Protocol, 2005, http://de.wikipedia.org/wiki/HTTP, Stand: 21.03.2005

[WIKIPDA] : Wikipedia, Personal Digital Assistant, 2005, http://de.wikipedia.org/wiki/Personal_Digital_Assistant, Stand: 25.03.2005

[PALOST5] : palmOne Inc., Tungsten T5 - Online Benutzerhandbuch, 2005, http://www.palmandmore.de/Produkte/fo_produkt.asp?P_Nr=2181&Ident=0031803200522010075013, Stand: 28.03.2005

[WIKIHANDY] : Wikipedia, Mobiltelefon, 2005, http://de.wikipedia.org/wiki/Handy, Stand: 25.03.2005

[U15SIM] : Siemens AG, Siemens Mobile - Be inspired - U15 - Benutzerhandbuch, 2004, http://communications.siemens.com/cds/frontdoor/0,2241,de_de_0_15812_rArNrNrNrN,00.html, Stand: 25.03.2005

[WIKISMART] : Wikipedia, Smartphone, 2005, http://de.wikipedia.org/wiki/Smartphone, Stand: 26.03.2005

[S80AT] : s80.at, Communicator-Workshop: Kalender, 2005, http://www.s60.at/communicator/forum/kalender.htm, Stand: 23.03.2005

[WIKIMIME] : Wikipedia, MIME, 2005, http://de.wikipedia.org/wiki/MIME, Stand: 13.04.2005

[WIKIICAL] : Wikipedia, iCalendar, 2005, http://de.wikipedia.org/wiki/ICalendar, Stand: 13.04.2005

[WIKIICALE] : Wikipedia, iCalendar, 2005, http://en.wikipedia.org/wiki/ICalendar, Stand: 13.04.2005

[WIKIUTC] : Wikipedia, Koordinierte Weltzeit, 2005, http://de.wikipedia.org/wiki/Koordinierte_Weltzeit, Stand: 13.04.2005

[RFC2445] : Stenerson und Dawson, Internet Calendaring and Scheduling Core Object Specification (iCalendar), November 1998, http://www.ietf.org/rfc/rfc2445.txt, Stand: 13.04.2005

[WIKISYNC] : Wikipedia, SyncML, 2005, http://de.wikipedia.org/wiki/SyncML, Stand: 16.04.2005

[SP2G] : Space2Go, White Paper SyncML, 2002, http://www.space2go.de, Stand: 16.04.2005

[TECSYNC] : Klaus Manhart, Daten immer up to date, Juni 2003, http://www.tecchannel.de/internet/762, Stand: 16.04.2005

[WIKIDAV] : Wikipedia, WebDAV, 2005, http://de.wikipedia.org/wiki/WebDAV, Stand: 18.04.2005

[WIKICAL] : Wikipedia, CalDAV, 2005, http://en.wikipedia.org/wiki/CalDAV, Stand

18.04.2005
[IEEECAL] : IEEE Internet Computing (Jim Whitehead), Open Calendar Sharing and Scheduling with CalDAV, 2005, http://dsonline.computer.org/portal/site/dsonline/menuitem.9ed3d9924aeb0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/0504&file=w2sta.xml&xsl=article.xsl, Stand: 18.04.2005

## 9.2 Abbildungsverzeichnis

## 9.3 Quellen der Abbildungen

*(Sofern keine Quelle angegeben ist, wurden die die Abbildungen selbst erstellt.)*

Abb. 1
*http://www.absatzplus.com/produkte/200/29/29051203.jpg (04.03.2005)*

Abb. 2
*Logo entnommen aus der Anwendung Cross-Desktop (04.03.2005)*

Abb. 18
*http://img.dell.com/images/emea/specialitems/products/dimen/8400_front_cutout_215x210.jpg (18.03.2005)*

Abb. 19
*http://de.wikipedia.org/wiki/Bild:IBM_Thinkpad_R51.jpg (18.03.2005)*

*Abb. 25*
*http://www.palmandmore.de/Produktbilder/2181_max.jpg (25.03.2005)*

Abb. 26 - 29
*Tungsten T5 - Online Benutzerhandbuch (08.04.2005)*

Abb. 30
*http://de.wikipedia.org/wiki/Bild:Handys.jpg (08.04.2005)*

Abb. 31
*http://communications.siemens.com/repository/212/21245/U15_front_big.jpg (08.04.2005)*

Abb. 32, 33
*Siemens Mobile - Be inspired - U15 – Benutzerhandbuch (08.04.2005)*

Abb. 34
*http://www.chip.pl/arts/kti/images/960538_5d27fe849c.jpg (08.04.2005)*
*http://www.chip.de/ii/19323050_3aca9aecc6.jpg (08.04.2005)*

Abb. 35-38
*http://www.s60.at/communicator/forum/kalender.htm (08.04.2005)*

Abb. 41/42
*http://www.tecchannel.de/internet/762 (14.04.2005)*

Abb. 43
*http://dsonline.computer.org/portal/cms_docs_dsonline/dsonline/w2sta01.gif (14.04.2005)*

Abb. 57
*http://upload.wikimedia.org/wikipedia/en/0/03/Mysql.jpg (20.04.2005)*

Abb. 58
*http://search.cpan.org/s/img/cpan_banner.png (20.04.2005)*